

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

**DESARROLLO DE UN SISTEMA PROTOTIPO DE ACCESO A LOS
LABORATORIOS DE REDES DE LA FACULTAD DE INGENIERÍA
ELÉCTRICA Y ELECTRÓNICA (FIEE) DE LA ESCUELA
POLITÉCNICA NACIONAL (EPN) BASADO EN RECONOCIMIENTO
FACIAL**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO ELECTRÓNICA Y REDES DE INFORMACIÓN**

CHRISTIAN XAVIER ARROYO AUZ

DIRECTOR: M.Sc. XAVIER ALEXANDER CALDERÓN HINOJOSA

Quito, septiembre 2019

AVAL

Certifico que el presente trabajo fue desarrollado por Arroyo Auz Christian Xavier, bajo mi supervisión.

M.Sc. CALDERÓN HINOJOSA XAVIER ALEXANDER
DIRECTOR DE TRABAJO DE TITULACIÓN

DECLARACIÓN DE AUTORÍA

Yo Arroyo Auz Christian Xavier, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración dejo constancia de que la Escuela Politécnica Nacional podrá hacer uso del presente trabajo según los términos estipulados en la Ley, Reglamentos y Normas vigentes.

ARROYO AUZ CHRISTIAN XAVIER

DEDICATORIA

“Imposible es sólo una palabra que usan los hombres débiles para vivir fácilmente en el mundo que se les dio, sin atreverse a explorar el poder que tienen para cambiarlo. Imposible no es un hecho, es una opinión. Imposible no es una declaración, es un reto. Imposible es potencial. Imposible es Temporal, Imposible no es nada.” Muhammad Ali.

A Dios.

Quien con su espíritu me dio la fuerza suficiente para la terminación de esta etapa de estudios; por haberme dado la vitalidad y la salud que permitió convertirme en un profesional y finalmente por haberme dado una gran familia.

A mi madre Elsy.

Por darme la vida, amarme, creer en mí, por tu sacrificio en tierras lejanas y por haber sacado lo mejor de mí. Siempre tendré en mi memoria todas las oraciones que, ante Dios, hiciste por mí. Siempre estas allí cuando necesito un abrazo, tu dulce y amable corazón sabe entender cuando necesito una amiga, tus bellos ojos son severos cuando necesito una lección. Gracias por apoyarme con tu trabajo y por ser una mujer que me hace llenar de orgullo.

A mi padre Aníbal.

Por darme la vida, amarme y creer en mí. Este trabajo es un peldaño más que he logrado cumplir y sin lugar a dudas es gracias a tu trabajo y al esfuerzo que has hecho por mí; la verdad, no sé qué sería de mi vida sin tu apoyo. Eres una de las pocas personas del mundo que siempre está allí, de forma incondicional. Si te rechazo, me perdonas. Si me equivoco, me acoges. Si estoy feliz, celebras conmigo. Si estoy triste, no sonrías hasta que yo sonría. Gracias por tu compañía y por tu amor.

A mi hermana Karina.

¿Volvería a compartir vientre junto a ti? No tengo que pensarlo dos veces ya que la respuesta es Sí. Siempre me pregunto ¿Cómo hacen las personas que no tiene hermana?

porque desde que tengo memoria has sido mi mayor felicidad y la vida sin ti sería un camino errante, sin esperanzas y sin ilusiones.

A mi hermano Giovanny.

Existe ocasiones en las que tener un hermano es mucho mejor que tener a un súper héroe a tu lado, tú has sido una parte muy importante para complementar mi felicidad, has sido la persona con quien más he llorado, con quien más me he enfadado y discutido; pero también me he dado cuenta que no podría vivir sin ti ya que eres mi mejor amigo.

A mi ñaña Tere.

Por ser una gran mujer y ser como una madre para mí. Tuve la tristeza de no tener cerca de mi madre, cuando era muy joven, pero Dios fue misericordioso conmigo y no me dejó desamparado; ya que te puso ti a mi lado para que con tu amor y dulzura trataras de suplir ese vacío. No tengo palabras que expresen mi gratitud y no sé cuál hubiera sido mi destino si no hubieras estado allí. Finalmente, agradezco a su esposo, Gustavo Viteri, ya que sin su apoyo no se habría podido lograr llegar a esta etapa de mi vida.

A mi mami Angelini.

Eres una persona con plata en tu cabello y oro en tu corazón. Uno de los placeres más satisfactorios que he tenido en mi vida es tenerte a ti; ya que tus cuidados y cariños cuando fui niño han sido parte fundamental de la persona que soy hoy. Cuando te miro a los ojos, veo el hermoso reflejo de toda una vida y me acuerdo de tus innumerables actos de amor mostrados. Y con todo el corazón te doy las gracias; porque para mí eres una mamá más que me crio con todo su corazón.

A mi papi Mel (QEPM).

Aparte de ser mi abuelo y padre, fuiste mi consejero y mi amigo. Sé que sigues y guías mis pasos desde allí arriba. Te recuerdo y se llena mi alma de gozo por todo lo que me diste mientras estuviste junto a mí, fuiste una deliciosa mezcla de historias y sonrisas, tenías la sabiduría de un búho y el corazón de un ángel. Ahora sé que el abrazo de padre dura más tiempo cuando él se va, ¡Te extraño viejito!

A mis familiares.

Andrés y Carolina que han sido como hermanos; porque cada uno de ustedes han aportado grandes y valiosas cosas en mi vida, me han ayudado a ser una mejor persona. A mi cuñado Esteban que se ha convertido en un gran amigo.

A mis amigos.

Carlos A, Cristina V, Yomara G, Diego G, Luis M, Johan M, Alejandro M, José C, Belén C, Tex A, Javier B, Benjamín P, David M, David P, Carla Z, Marco C y Soraya A, por haber compartido los buenos y malos momentos junto a mí y a todos aquellos amigos que no he recordado al escribir estas palabras.

Con todo mi cariño, Christian Arroyo.

AGRADECIMIENTOS

A mi director, el M.Sc. Xavier Calderón por ser una gran persona y llevarme a enfocarme a realizar este trabajo, por su apoyo en mi vida estudiantil y en las prácticas pre profesionales. Al M.Sc. Pablo Hidalgo por darme la apertura a desarrollar este trabajo. Y finalmente a todos los docentes de la Escuela Politécnica Nacional que me prepararon para afrontar los retos venideros de la vida.

Con todo mi aprecio, Christian Arroyo.

ÍNDICE DE CONTENIDO

AVAL	I
DECLARACIÓN DE AUTORÍA.....	II
DEDICATORIA	III
AGRADECIMIENTOS	VI
ÍNDICE DE CONTENIDO	VII
ÍNDICE DE FIGURAS	X
ÍNDICE DE TABLAS	XV
ÍNDICE DE CÓDIGOS	XVI
GLOSARIO DE ABREVIATURAS	XVII
RESUMEN	XIX
ABSTRACT	XX
1. INTRODUCCIÓN.....	1
1.1. OBJETIVOS.....	2
1.2. ALCANCE	3
1.3. MARCO TEÓRICO	6
1.3.1. MICROSOFT VISUAL STUDIO	6
1.3.1.1. Visual Studio 2017.....	7
1.3.1.2. ¿Qué es Microsoft .Net Framework?.....	8
1.3.1.3. Windows Communication Foundation	11
1.3.1.4. C Sharp	13
1.3.2. LINQ A SQL.....	15
1.3.2.1. LINQ.....	16
1.3.2.2. SQL.....	18
1.3.3. OPENCV.....	18
1.3.4. EMGUCV	20
1.3.5. OZEKI CAMERA SDK.....	22
1.3.6. BIOMÉTRICOS	23
1.3.6.1. Etapas de un sistema biométrico	23
1.3.6.2. Rendimiento de un sistema biométrico	26
1.3.6.3. Requisitos de un sistema biométrico	28

1.3.6.4. Ventajas y desventajas de los sistemas biométricos	28
1.3.7. DETECCIÓN Y RECONOCIMIENTO FACIAL	29
1.3.7.1. Detección facial	30
1.3.7.2. Reconocimiento facial.....	33
1.3.8. HUELLA DACTILAR	41
1.3.9. METODOLOGÍAS DE DESARROLLO DE SOFTWARE	45
1.3.10. METODOLOGÍA DE DESARROLLO SOFTWARE KANBAN	45
1.3.11. SOFTWARE Y HARDWARE	46
2. METODOLOGÍA.....	49
2.1. DISEÑO DEL SISTEMA PROTOTIPO	49
2.1.1. SELECCIÓN METODOLOGÍA DE DESARROLLO SOFTWARE KANBAN.....	49
2.1.2. SITUACIÓN ACTUAL Y LAS NECESIDADES PRESENTES EN EL LABORATORIO DE REDES DE LA FIEE DE LA EPN	51
2.1.3. REQUERIMIENTOS FUNCIONALES Y NO FUNCIONALES	51
2.1.4. DIAGRAMAS DE CASOS DE USOS Y SECUENCIA PARA EL USO DE LA APLICACIÓN DEL SISTEMA PROTOTIPO	54
2.1.5. DIAGRAMAS DE CLASE DE LOS MÓDULOS: INTERFAZ DE USUARIO, LÓGICA DE NEGOCIOS Y ACCESO A DATOS	66
2.1.6. DIAGRAMAS DE ACTIVIDADES DE CADA MÓDULO: INTERFAZ DE USUARIO, LÓGICA DE NEGOCIOS Y ACCESO A DATOS	70
2.1.7. DIAGRAMAS ENTIDAD RELACIÓN DEL MÓDULO BASE DE DATOS	77
2.1.8. REQUERIMIENTOS FÍSICOS DEL SISTEMA DE SEGURIDAD....	77
2.2. IMPLEMENTACIÓN DEL SISTEMA PROTOTIPO	79
2.2.1. INSTALACIÓN DE VISUAL STUDIO 2017	80
2.2.2. INSTALACIÓN LIBRERÍAS OZEKI, EMGUCV, OPENCV, ZKTECO E ITEXT	82
2.2.3. CREACIÓN DE LOS FORMULARIOS DE LA INTERFAZ GRÁFICA.....	89
2.2.4. IMPLEMENTACIÓN DE CADA MÓDULO	99
2.2.5. CONFIGURACIÓN DE LOS TERMINALES DE ACCESO	110
2.2.6. CONFIGURACIONES DE EQUIPOS INTERMEDIOS	111

2.2.7. UBICACIÓN FÍSICA DE LOS TERMINALES DE ACCESO	113
2.2.8. CABLEADO PARA LA COMUNICACIÓN ENTRE LOS EQUIPOS .	116
3. RESULTADOS Y DISCUSIÓN.....	120
3.1. COMPROBACIÓN Y CORRECCIÓN DEL FUNCIONAMIENTO DE LOS MÓDULOS	120
3.2. VERIFICAR Y DEPURAR LA EXISTENCIA DE ERRORES PRODUCIDOS AL INTERCONECTAR LOS EQUIPOS	132
3.3. FUNCIONAMIENTO DE LAS CÁMARAS, LECTORES DE HUELLAS Y PINES DE ACCESO NUMÉRICO	132
3.4. ENCUESTAS DE SATISFACCIÓN	144
4. CONCLUSIONES Y RECOMENDACIONES.....	146
4.1. CONCLUSIONES	146
4.2. RECOMENDACIONES	147
5. REFERENCIAS BIBLIOGRÁFICAS	149
ANEXOS	155

ÍNDICE DE FIGURAS

CAPÍTULO 1

Figura 1.1. Prototipo del sistema de seguridad basado en reconocimiento facial.	3
Figura 1.2. Módulos del prototipo de sistema de seguridad.	4
Figura 1.3. Detección de un rostro por la cámara.	5
Figura 1.4. Detección de rostro y almacenamiento/comparación en la base de datos.	5
Figura 1.5. Comunicación entre los componentes de una aplicación .NET.	9
Figura 1.6. Entorno común de ejecución para lenguajes.	10
Figura 1.7. Biblioteca de clases base.	10
Figura 1.8. Tecnología combinada para formar WCF.	12
Figura 1.9. Compilación y creación del archivo ejecutable en C#.	14
Figura 1.10. Mapeador de objetos relacionales.	16
Figura 1.11. Arquitectura LINQ.	17
Figura 1.12. Representación y modificación de una imagen mediante el uso de OpenCV.	20
Figura 1.13. Requisitos básicos para crear una aplicación con Ozeki SDK.	22
Figura 1.14. Etapas en un sistema de identificación biométrica.	23
Figura 1.15. Identificación (arriba) y autenticación (abajo), diferencias entre las fases.	25
Figura 1.16. Rendimiento de un sistema biométrico resumido en un gráfico DET.	27
Figura 1.17. Sensibilidad del sistema biométrico.	27
Figura 1.18. Cambio de imagen original a una imagen integral.	31
Figura 1.19. Características haar.	32
Figura 1.20. Las dos características de haar más significativas seleccionadas por AdaBoost.	33
Figura 1.21. Procedimiento para el reconocimiento facial.	34
Figura 1.22. Puntos duros en un rostro.	34
Figura 1.23. Diferentes pixelados en una imagen.	35
Figura 1.24. Técnicas del reconocimiento facial.	35

Figura 1.25. Imágenes de entrenamiento.....	37
Figura 1.26. Conjunto de EigenFaces.....	37
Figura 1.27. Cuadro Fisher de una imagen de entrada.....	38
Figura 1.28. Operación LBP aplicada a una imagen.....	39
Figura 1.29. Imagen orinal (izquierda) e Imagen procesada con LBP (derecha). ..	40
Figura 1.30. Algunas de las características de una huella dactilar.....	42
Figura 1.31. Diferentes tipos de minucias.....	42
Figura 1.32. Diferentes tipos de patrón de huellas dactilares.....	43
Figura 1.33. Procesamiento de una huella dactilar.....	44
Figura 1.34. Tablero Kanban.....	46

CAPÍTULO 2

Figura 2.1. Diagrama de caso de uso general.....	54
Figura 2.2. Diagrama de caso de uso del inicio de sesión.....	55
Figura 2.3. Diagrama de secuencia del inicio de sesión.....	56
Figura 2.4. Diagrama de caso de uso del CRUD de usuarios.....	57
Figura 2.5. Diagrama de secuencia del registro y modificación de usuarios.....	58
Figura 2.6. Diagrama de secuencia de la eliminación de usuarios.....	59
Figura 2.7. Diagrama de caso de uso administración del log de eventos.....	60
Figura 2.8. Diagrama de secuencia del log de eventos.....	61
Figura 2.9. Diagrama de caso de uso para la administración de dispositivos.....	62
Figura 2.10. Diagrama de secuencia para la administración de dispositivos.....	63
Figura 2.11. Diagrama de caso de uso para la manipulación de las imágenes de video.....	64
Figura 2.12. Diagrama de secuencia para la manipulación de las imágenes de video.....	65
Figura 2.13. Diagrama de clases de la interfaz de usuario y la lógica de negocios.....	66
Figura 2.14. Diagrama de clases de la interfaz de usuario.....	67
Figura 2.15. Diagrama de clases lógica de negocios.....	68
Figura 2.16. Diagrama de clases del acceso a datos y servidor.....	69
Figura 2.17. Diagrama de clases del acceso a datos y servidor.....	70

Figura 2.18. Diagrama de actividades del login.	71
Figura 2.19. Diagrama de actividades del CRUD usuarios.	72
Figura 2.20. Diagrama de actividades del log de eventos.	73
Figura 2.21. Diagrama de actividades de la administración de dispositivos.	74
Figura 2.22. Diagrama de actividades de la manipulación de las imágenes de Video.....	75
Figura 2.23. Diagrama de actividades lógica de negocios y acceso a datos.	76
Figura 2.24. Diagrama entidad relación del módulo base de datos.	77
Figura 2.25. Página para descarga de Visual Studio.	80
Figura 2.26. Parámetros posibles a instalar en Visual Studio.	81
Figura 2.27. Pantalla de inicio de Visual Studio.	81
Figura 2.28. Descarga librería OpenCV.	82
Figura 2.29. Descarga librería EmguCV.	82
Figura 2.30. Ruta de extracción archivos OpenCV.	83
Figura 2.31. Carpeta creada con las librerías de OpenCV.	83
Figura 2.32. Ruta instalación archivo EmguCV.	84
Figura 2.33. Depurador de EmguCV en Visual Studio.	84
Figura 2.34. Descarga del SDK de Ozeki.	85
Figura 2.35. Ruta de instalación archivos Ozeki.	86
Figura 2.36. Interfaz del Ozeki Cámara SDK.	86
Figura 2.37. Descarga SDK ZKTeco.	87
Figura 2.38. Instalación SDK ZKTeco.	88
Figura 2.39. Finalización instalación SDK ZKTeco.	88
Figura 2.40. Descarga librería Interop.ZKFPEngXControl.dll.	89
Figura 2.41. Creación de un nuevo proyecto en Visual Studio.	90
Figura 2.42. Creación de solución en blanco del nuevo proyecto.	91
Figura 2.43. Creación de nuevo proyecto dentro de la solución.	92
Figura 2.44. Creación nuevo formulario en el proyecto.	93
Figura 2.45. Interfaz gráfica del formulario principal.	94
Figura 2.46. Interfaz gráfica del log de eventos.	95
Figura 2.47. Interfaz gráfica de la administración de usuarios.	96
Figura 2.48. Interfaz gráfica para agregar lectores.	97
Figura 2.49. Interfaz gráfica para agregar teclados.	98

Figura 2.50. Interfaz gráfica del inicio de sesión.	99
Figura 2.51. Creación de la base de datos.	105
Figura 2.52. Prueba de conexión a la base de datos.	106
Figura 2.53. Configuración de la cámara de video.	111
Figura 2.54. Configuración del router.....	112
Figura 2.55. Ubicación física de los terminales de acceso en la pared.	114
Figura 2.56. Vista externa caja electrónica.	115
Figura 2.57. Vista interna caja electrónica.....	115
Figura 2.58. Implementación del sistema de seguridad.	116
Figura 2.59. Interruptor para abrir la cerradura internamente.....	117
Figura 2.60. Cableado hacia el techo raso.	117
Figura 2.61. Cerradura en la puerta.....	118
Figura 2.62. Acercamiento a la cerradura.	118

CAPÍTULO 3

Figura 3.1. Validación de un textbox.	121
Figura 3.2. Validación selección de una cámara.	122
Figura 3.3. Acceso y búsqueda en la datos y base de datos.	124
Figura 3.4. Documento impreso en el formato especificado.	125
Figura 3.5. Incompatibilidad con la resolución de pantalla.	126
Figura 3.6. Corrección de incompatibilidad con la resolución de pantalla.	127
Figura 3.7. Base de datos incompatible.	128
Figura 3.8. Bases de datos de SQL Server.	128
Figura 3.9. Migrando la base de datos del proyecto a SQL Server.....	129
Figura 3.10. Conexión de la base de datos en el nuevo computador.....	130
Figura 3.11. Actualizando la cadena de conexión de la base de datos.	131
Figura 3.12. Fallo en la conexión con el Arduino.	132
Figura 3.13. Comprobación del funcionamiento de la cámara.....	134
Figura 3.14. Cámara en red inalcanzable.....	135
Figura 3.15. Imagen con malas condiciones de iluminación.	136
Figura 3.16. Imagen con malas condiciones de altura.....	137
Figura 3.17. Imagen con mal ángulo de observación.	138

Figura 3.18. Funcionamiento del lector dactilar.	139
Figura 3.19. Funcionamiento del teclado numérico.	140
Figura 3.20. Resultados del funcionamiento de la aplicación.	141
Figura 3.21. Librerías utilizadas en el proyecto.....	143

ÍNDICE DE TABLAS

CAPÍTULO 1

Tabla 1.1. Frameworks compatibles con Visual Studio 2017.	11
Tabla 1.2. Tipos de datos, básicos, en la biblioteca de clases de .NET.	15
Tabla 1.3. Versiones de EmguCV.....	21
Tabla 1.4. Comparación entre algunos métodos biométricos.....	29
Tabla 1.5. Metodologías para el desarrollo de software.	45
Tabla 1.6. Elementos hardware a utilizar.	47
Tabla 1.7. Elementos software a utilizar	48

CAPÍTULO 2

Tabla 2.1. Tablero Kanban fase inicial de diseño.....	50
Tabla 2.2. Requerimientos funcionales.	52
Tabla 2.3. Requerimientos no funcionales.	52
Tabla 2.4. Primera actualización tablero Kanban.	53
Tabla 2.5. Requerimientos físicos del sistema de seguridad.....	78
Tabla 2.6. Segunda Actualización Tablero Kanban.	79
Tabla 2.7. Tercera actualización tablero Kanban.	119

CAPÍTULO 3

Tabla 3.1. Estadísticas de las pruebas de funcionamiento.	142
Tabla 3.2. Resultados de las encuestas de satisfacción.	144
Tabla 3.3. Resultados de las encuestas de satisfacción (continuación).	145
Tabla 3.4. Cuarta Actualización Tablero Kanban	145

ÍNDICE DE CÓDIGOS

CAPÍTULO 1

Código 1.1. Consulta SQL sin usar ORM a una base de datos.....	17
Código 1.2. Consulta LINQ usando ORM a una base de datos.	17

CAPÍTULO 2

Código 2.1. Método para bloquear los controles visuales.	100
Código 2.2. Anulación del evento cerrar.	100
Código 2.3. Conexión con el servidor.	101
Código 2.4. Reconocimiento de teclados y procesamiento de mensajes.	101
Código 2.5. Detección y reconocimiento facial.	102
Código 2.6. Detección y reconocimiento dactilar.	103
Código 2.7. Registrar un usuario en la base de datos.	103
Código 2.8. Iniciar el servidor.	104
Código 2.9. Creación de la tabla Usuarios.	107
Código 2.10. Creación de la tabla Usuarios en Visual Studio.	107
Código 2.11. Cadena de conexión a la base de datos.	108
Código 2.12. Agregación de los servicios....	108
Código 2.13. Consultas LINQ.....	109
Código 2.14. Agregar un usuario mediante LINQ.	110
Código 2.15. Programación del Arduino.....	113

GLOSARIO DE ABREVIATURAS

ADO	ActiveX Data Objects (Objetos de Datos ActiveX).
API	Application Programming Interface (Interfaz de Programación de Aplicaciones).
BCL	Base Class Library (Biblioteca de Clases Base).
BSD	Berkeley Software Distribution (Distribución de Software Berkeley).
CCD	Charge-Coupled Device (Dispositivo de Carga Acoplada).
CLR	Common Language Runtime (Entorno en Tiempo de Ejecución de Lenguaje Común).
CRUD	Create, Read, Update and Delete (Crear, Leer, Actualizar y Borrar).
C#	C Sharp.
DCL	Data Control Language (Lenguaje de Control de Datos).
DDL	Data Definition Language (Lenguaje de Definición de Datos).
DET	Detection Error Tradeoff (Gráfica de Compensación de Error).
DHCP	Dynamic Host Configuration Protocol (Protocolo de Configuración Dinámica de Host).
DML	Data Manipulation Language (Lenguaje de Manipulación de Datos).
DNI	Documentos Nacionales de Identidad.
EPN	Escuela Politécnica Nacional.
FAR	False Acceptance Rate (Tasa de Falsa Aceptación).
FER	Failure-to-enroll Rate (Tasa de Error de Reclutamiento).
FIEE	Facultad Ingeniería Eléctrica y Electrónica.
FLD	Fisher's linear discriminant (Discriminante Lineal de Fisher).
FNMR	False NonMatch Rate (Tasa de No Coincidencia Falsa).
FRR	False Rejection Rate (Tasa de Falso Rechazo).
FTE	Failure-to-enroll Rate (Tasa de Error de Reclutamiento).
GPL	GNU General Public License (Licencia Pública General de GNU).
GNU	GNU's Not Unix (del Acrónimo: GNU no es Unix).
GUI	Graphical User Interface (Interfaz Gráfica de Usuario).
IDE	Integrated Development Environment (Entorno de Desarrollo Integrado).
IL	Intermediate Language (Lenguaje Intermedio).
IP	Internet Protocol (Protocolo de Internet).
LAN	Local Area Network (Redes De Área Local).

LBPH	Local Binary Patterns Histograms (Histogramas de Patrones Binarios Locales).
LINQ	Language-Integrated Query (Lenguaje Integrado de Consultas).
MBPS	Megabit Per Second (Megabit Por Segundo).
MSF	Microsoft Solutions Framework (Marco de soluciones de Microsoft).
ONVIF	Open Network Video Interface Forum (Foro de Interfaz de Video en Red Abierta).
ORM	Object-Relational Mapping (Mapeador de Objetos Relacionales).
PCA	Principal Component Analysis (Análisis de Componentes Principales).
PIN	Personal Identification Number (Número de Identificación Personal).
SDK	Software Development Kit (Kit de Desarrollo de Software).
SQL	Structured Query Language (Lenguaje de Consulta Estructurada).
RTSP	Real Time Streaming Protocol (Protocolo de Transmisión en Tiempo Real).
RUP	Rational Unified Process (Proceso Racional Unificado).
USB	Universal Serial Bus (Bus Universal en Serie).
VB	Visual Basic.
VSTU	Visual Studio Tools for Unity (Herramientas de Visual Studio para Unity).
WAN	Wide Area Network (Redes De Área Amplia.).
WCF	Windows Communication Foundation (Fundación de comunicación de Windows o también conocido como Índigo).

RESUMEN

Este trabajo de titulación propone implementar un sistema prototípico de acceso al Laboratorio de Redes de la Facultad de Ingeniería Eléctrica y Electrónica (FIEE) de la Escuela Politécnica Nacional (EPN) basado en reconocimiento facial; para lo cual se utiliza el entorno de desarrollo integrado “Visual Studio” con el lenguaje de programación orientado a objetos “C#” para la aplicación de escritorio en conjunto con las librerías OpenCV, EmguCV y Ozeki sobre un ambiente Windows. Como hardware se utiliza: cámara IP, lector de huella dactilar, panel numérico, router, usb hub y un computador donde se alojará la aplicación de escritorio.

En el capítulo uno, se expone la problemática de control de acceso que presenta el Laboratorio de Redes de la FIEE de la EPN y las alternativas de solución al problema. Adicionalmente, se detallan los aspectos teóricos sobre el software y las librerías que se utilizan para el desarrollo de la aplicación. Finalmente, se describe el funcionamiento, de algunos, de los algoritmos para la detección de rostros y huellas digitales, así como los equipos hardware que se utilizan.

En el capítulo dos, se presentan los requerimientos del sistema prototípico a implementar, tanto funcionales como no funcionales. Se detallarán los procesos de instalación del software y librerías utilizadas, así como la creación misma de la aplicación de escritorio.

El capítulo tres, se muestra las pruebas de funcionamiento del sistema prototípico de seguridad.

El capítulo cuatro, se presentan las conclusiones y recomendaciones que se logró obtener a lo largo del desarrollo de este proyecto.

PALABRAS CLAVE: Detección de rostros, Biométricos, Visual Studio, OpenCV, EmguCV y Ozeki.

ABSTRACT

This titling work proposes a system of access to the Network Laboratory of the Faculty of Electrical and Electronic Engineering (FIEE) of the National Polytechnic School (EPN) based in facial recognition; for which the integrated development environment "Visual Studio" with the Object-oriented programming language "C#" for desktop application is used in conjunction with the OpenCV, EmguCV and Ozeki libraries on a Windows environment. As hardware is used: IP camera, fingerprint reader, numeric panel, router, usb hub and a computer where the desktop application will be housed.

In chapter one, the problem of access control presented by the Laboratory of Networks of the FIEE of the National Polytechnic School and alternative solutions to the problem are exposed. In addition, the theoretical aspects of the software and the libraries that are used for the development of the desktop application are detailed. Finally, describe the operation of some of the algorithms for the detection of faces and fingerprints, as well as the equipment that is used.

In chapter two, the requirements of the prototype system to be implemented, both functional and non-functional, are presented. The installation processes of the software and libraries used will be detailed, as well as the creation of the desktop application itself.

Chapter three shows the operating tests of the prototype safety system.

Chapter four presents the conclusions and recommendations that are obtained throughout the development of this project.

KEYWORDS: Face Detection, Biometrics, Visual Studio, OpenCV, EmguCV and Ozeki.

1. INTRODUCCIÓN

El presente trabajo de titulación propone resolver el problema de control de acceso sobre el Laboratorio de Redes de la FIEE de la EPN, ya que no cuenta con un sistema de seguridad electrónico para el acceso a las diferentes salas del Laboratorio de Redes; la integración de un sistema de seguridad electrónico ayudaría a mejorar la seguridad, ya que este sistema contará con control de acceso biométrico y cámara de seguridad que permitirán reconocer a cada persona que ingrese.

Debido a este problema se propone como solución el “Desarrollo de un sistema prototípico de acceso al Laboratorio de Redes de la Facultad de Ingeniería Eléctrica y Electrónica de la Escuela Politécnica Nacional basado en reconocimiento facial”.

En los últimos años el uso de biométricos, como la detección facial, como sistemas de seguridad ha ido creciendo; ya que este permite extraer características físicas de las personas como lo son los ojos, nariz, boca, cejas, etc.

A lo largo del mundo diferentes empresas, compañías e incluso personas comunes hacen uso de la tecnología de detección de rostros; esta tecnología permite acceder a diferentes lugares e incluso les permite obtener información personal. Esta misma tecnología es usada habitualmente en la telefonía móvil; ya que muchos celulares actuales proveen de este servicio para desbloquear el dispositivo.

El que exista esta tecnología y que sea aplicable, en varios campos de la vida cotidiana, indica que es posible aplicarlo para incrementar la seguridad de los equipos vulnerables del Laboratorio de Redes de la FIEE en la EPN.

A continuación, se presentan algunos ejemplos de uso de esta tecnología:

- Para evitar los problemas de inseguridad ciudadana, estudiantes de la Universidad Continental del Perú, en el XXIV Congreso y Exposición Internacional de Ingeniería Mecánica, Mecatrónica, Eléctrica, Electrónica y ramas afines (CONEIMERA) en la ciudad de Trujillo, desarrollaron un proyecto basado en la detección de rostros hecho en Matlab obteniendo resultados positivos tras sus pruebas con imágenes de documentos nacionales de identidad (DNI) [24].

- En Shanghái – China, se desarrolló la tecnología “Face++” la cual es una aplicación que permite reconocer rostros y realizar diferentes tipos de pagos como en restaurantes, centros comerciales, taxis, etc.; incluso permite la integración con aplicaciones de pago como “Alipay”, la más usada en China [25].
- En Quito – Ecuador, en la Escuela Politécnica Nacional se han desarrollado varios proyectos basados en el reconocimiento facial los cuales han sido implementados y se ha logrado obtener buenos resultados tras las pruebas de campo; estos fueron escritos en diferentes lenguajes de programación y haciendo uso de diferentes tecnologías como son: “Diseño e implementación de un sistema de seguridad basado en reconocimiento de rostros [1]” el cual fue escrito en LabView, “Reconocimiento de rostros utilizando redes neuronales [2]” el cual fue desarrollado en Matlab y finalmente, “Diseño e implementación de un sistema de seguridad y alerta para vehículos, basado en reconocimiento facial y localización GPS, en una Raspberry pi b plus [5]” el mismo que hace uso de la tecnología de placas reducidas de Raspberry.

El prototipo está orientado a tratar de mejorar el sistema de seguridad ya existente del Laboratorio de Redes de la FIEE de la EPN, donde se encuentran los diferentes tipos de equipos que son usados por estudiantes y profesores en sus actividades cotidianas y con el objetivo de evitar posibles pérdidas materiales y daños a las personas que allí laboran y estudian se propone proveer de un sistema de acceso a las salas de Laboratorio de Redes de la FIEE mediante el uso de equipos que permitan conceder el ingreso al personal calificado, con el cual se puede reducir los diferentes tipos de amenazas que pueden existir.

1.1. OBJETIVOS

Objetivo General: Desarrollar un sistema prototipo para el acceso a los Laboratorios de Redes de la FIEE de la EPN basado en reconocimiento facial.

Objetivos Específicos:

1. Describir los fundamentos teóricos y las herramientas de software requeridas para el desarrollo del sistema prototipo.
2. Diseñar los módulos del software del sistema prototipo.
3. Implementar el sistema de seguridad con cada uno de los módulos del software del sistema prototipo.
4. Analizar los resultados obtenidos.

1.2. ALCANCE

El sistema prototípico de seguridad del Laboratorio de Redes de la FIEE - EPN que se desarrollará constará de un sistema de control de acceso (rostro, huella digital y PIN de acceso), un sistema de interconexión que permita la comunicación de los equipos con los dispositivos intermedios y este a su vez con un computador donde se encuentre alojada la base de datos del sistema de seguridad y la respectiva aplicación desarrollada en C#.

En la Figura 1.1, se muestra cómo estará estructurado a nivel de hardware el sistema del prototípico.

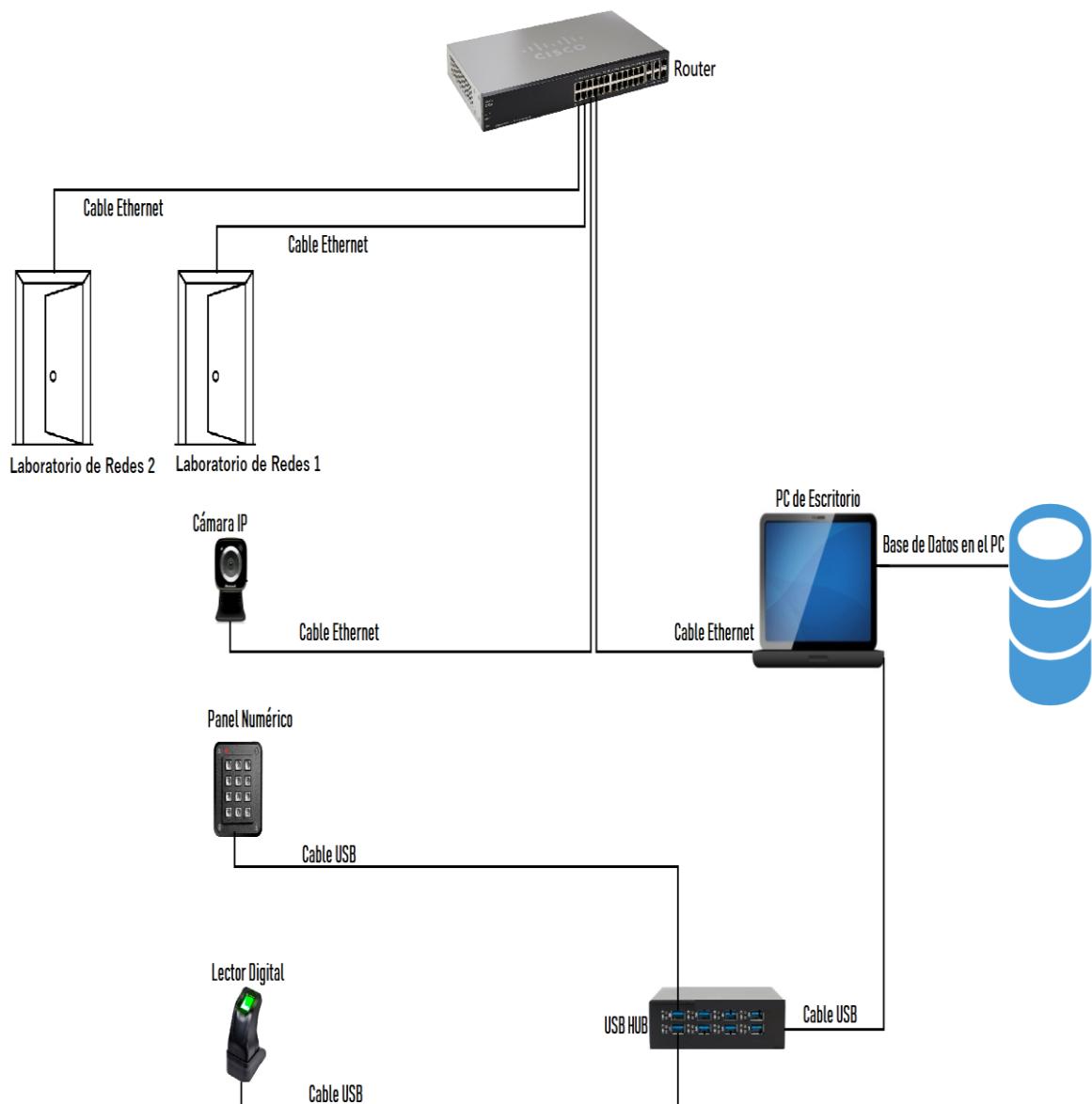


Figura 1.1. Prototipo del sistema de seguridad basado en reconocimiento facial.

La aplicación estará compuesta de varios módulos, como se muestra en la Figura 1.2, los mismos que permitirán realizar las tareas de administración y control del sistema prototipo de seguridad; estos módulos serán:

- Interfaz de Usuario: será el módulo mediante el cual los usuarios podrán interactuar con la aplicación.
- Lógica de Negocios: este módulo realizará la conexión entre el cliente y el servidor, permitirá reconocer a los dispositivos periféricos y determinar si estos están activos o no para su funcionamiento; adicionalmente, el módulo permitirá recibir los datos enviados por los dispositivos periféricos los cuales serán procesados; además este módulo se comunicará con el módulo de “Acceso a Datos” para solicitar al gestor de “Base de Datos” realizar consultas SQL.
- Acceso a Datos: proporcionará un acceso a los datos guardados en la “Base de Datos”; además permitirá realizar consultas SQL.
- Base de Datos: es donde se encuentran todas las tablas requeridas con toda la información para el correcto funcionamiento de la aplicación de escritorio.



Figura 1.2. Módulos del prototipo de sistema de seguridad [26].

El sistema prototipo permitirá que la cámara se encargue de tomar una fotografía de la persona que desee ingresar, como se muestra en la Figura 1.3, esta imagen será enviada hacia el computador por medio de los dispositivos intermedios, donde será procesada por uno de los módulos y retornará la imagen mostrando que el rostro ha sido reconocido.

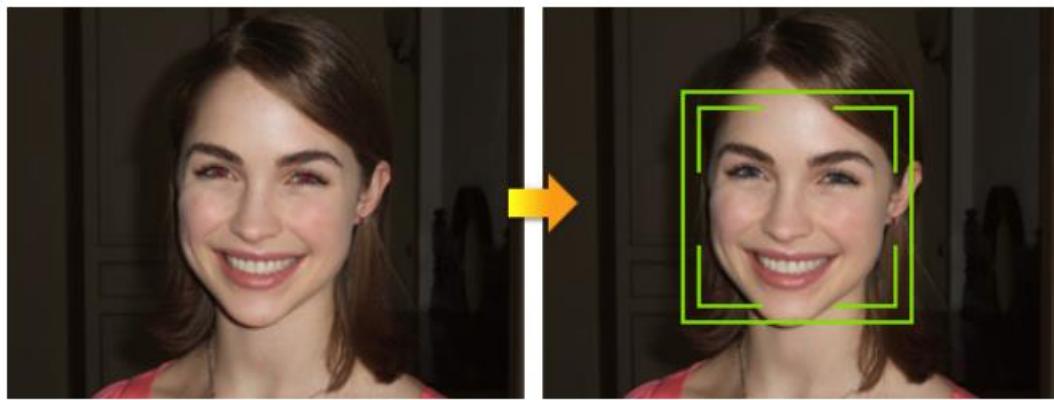


Figura 1.3. Detección de un rostro por la cámara [27].

Además, existirá un lector de huellas digitales y teclado numérico mediante el cual se podrá tener acceso al laboratorio en el caso de que el sistema principal, la detección del rostro, fallase.

Los equipos intermedios, que pueden ser desde un solo switch hasta varios equipos formando una red, serán los encargados de interconectar los equipos de control de acceso con el equipo donde se encuentre alojada la aplicación y su respectiva base de datos.

La base de datos, almacenará toda la información relacionada a todos los usuarios; esta irá desde información básica como: nombre, apellido, cédula, PIN de acceso, etc., hasta la información de los biométricos de la persona como: huella dactilar y patrón del rostro.

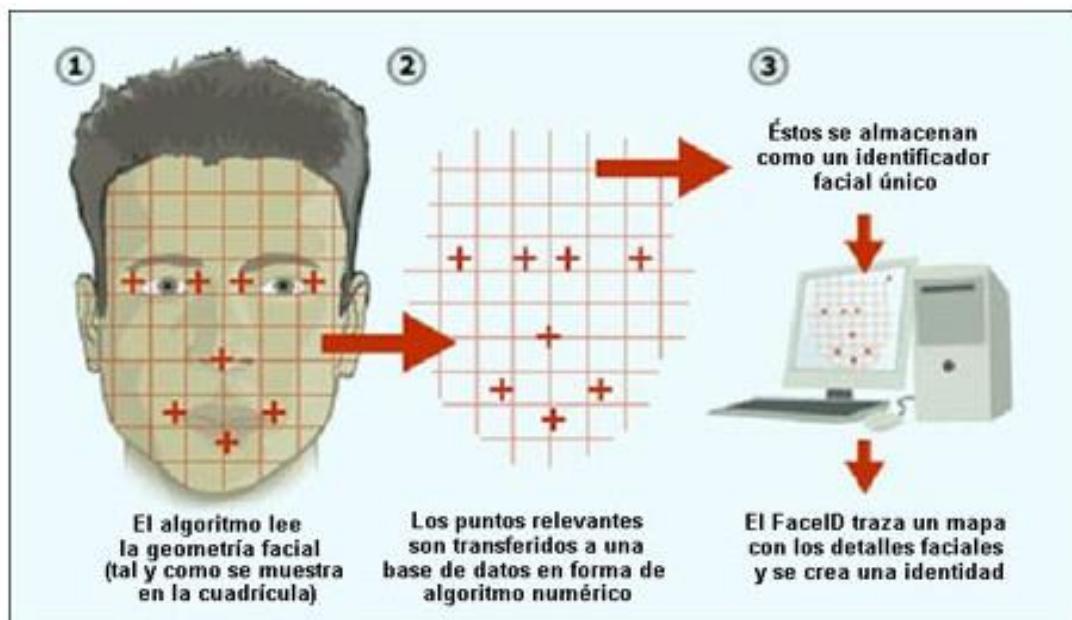


Figura 1.4. Detección de rostro y almacenamiento/comparación en la base de datos [23].

La aplicación de escritorio estará instalada en un computador; esta será la encargada de realizar el reconocimiento facial comparando el patrón de rostro obtenido con uno alojado en la base de datos, como se muestra en la Figura 1.4. Además, se encargará de dar el acceso o de denegarlo; también tendrá la función de dar acceso mediante el PIN de acceso o por medio de la huella digital si se diera el caso de que el mecanismo principal fallará.

Se escogerá una metodología de desarrollo de software adecuada para el desarrollo del sistema prototipo que permita implementar todos los requerimientos del sistema prototipo.

1.3. MARCO TEÓRICO

En esta etapa se recopilará toda la información necesaria sobre las herramientas software y hardware y se procederá a realizar el respectivo marco teórico que describa la información relevante sobre algoritmos de detección de rostros, lenguajes de programación, tipo de comunicación entre equipos, base de datos y aplicaciones de escritorio y su funcionalidad.

1.3.1. MICROSOFT VISUAL STUDIO

“Microsoft Visual Studio es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistemas operativos Windows. Soporta múltiples lenguajes de programación, tales como: C++, C#, Visual Basic .NET, F#, Java, Python, Ruby y PHP, al igual que entornos de desarrollo web, como ASP.NET MVC, Django, etc., a lo cual hay que sumarle las nuevas capacidades online bajo Windows Azure en forma del editor Mónaco [28]”.

Visual Studio es utilizada por los programadores, ya que proporciona un entorno rico e integrado para crear aplicaciones para Windows, Android e iOS; siempre y cuando su sistema operativo soporte la plataforma .NET; para ello Visual Studio usa plataformas de desarrollo de software de Microsoft como: Windows Forms, API de Windows, Windows Presentation, etc.; adicionalmente, cabe mencionar que Visual Studio provee de un diseñador de clases y un diseñador de esquema de base de datos.

El IDE es una aplicación informática que permite a los desarrolladores varios servicios que facilita el desarrollo de software; básicamente un IDE está compuesto por una serie de librerías y servicios que pueden ser manipuladas a través de una GUI, un editor de código fuente y un depurador; en ciertos casos el IDE puede estar compuesto por un compilador,

un intérprete y un auto-completado inteligente de código. “Visual Studio incluye un editor de código compatible con IntelliSense [29]” (auto-completado inteligente de código), así como un depurador.

1.3.1.1. Visual Studio 2017

Visual Studio 2017 ofrece a sus usuarios una serie de mejoras, con respecto a sus predecesores, las mismas que se enfocan al desarrollo de aplicaciones móviles, aplicaciones web, aplicaciones en la nube, desarrollo Azure, entre otras; entre las nuevas características que ofrece se puede mencionar: conjunto de herramientas .NET Core, editor XAML, IntelliSense, pruebas de unidad en vivo, mejora de depuración, etc.

Este es un resumen general de los aspectos más importantes de Visual Studio 2017. Aspectos obtenidos de “Microsoft Docs – Novedades de Visual Studio 2017 [30]”:

- Redefinición de los aspectos básicos: el proceso de instalación es más rápido e intuitivo, ya que cuenta con una interfaz más sencilla donde se puede escoger, de una lista, los aspectos que el usuario necesite.
- Rendimiento y productividad: implementa nuevas funciones de desarrollo en la nube y aplicaciones móviles. Adicionalmente, tiene mayor capacidad de respuesta y menor requerimiento en el uso de memoria.
- Desarrollo de aplicaciones en la nube con Azure: permite la creación de aplicaciones destinadas a la nube, facilitando la configuración, la compilación, la depuración, el empaquetado y la implementación de aplicaciones y servicios.
- Desarrollo de aplicaciones para Windows: permite la creación de aplicaciones para una amplia gama de dispositivos Windows como: computadoras con sistema operativo Windows 10, tabletas, teléfonos, Xbox y más.
- Desarrollo de juegos: con VSTU, puede usar Visual Studio para escribir scripts de editor y juegos en C#.
- Desarrollo multiplataforma: con Visual Studio es posible distribuir software (aplicaciones y librerías) que se ejecuten en sistemas operativos Windows, Linux y macOS sin problemas.

En el “Anexo A”, “Requisitos del sistema de Visual Studio 2017 [31]”, se muestran los requerimientos mínimos para el uso de Visual Studio 2017, en sus versiones Profesional, Enterprise Community, Team Explore, Team Community, Test Agent, Test Controller, Feedback Client y Team Foundation Server Office Integration.

1.3.1.2. ¿Qué es Microsoft .Net Framework?

Para aquellos usuarios de Windows, es probable, que revisando las características del sistema operativo se hayan encontrado con programas instalados de “.NET Framework” en diferentes versiones y se han preguntado qué funcionalidad tiene ese programa. “.NET es una plataforma de ejecución de Aplicaciones basada en objetos y desarrollada por la empresa Microsoft destinada a facilitar la vida de los desarrolladores de aplicaciones (en especial a los usuarios de Visual Basic y Visual Interdev) [15]”.

.NET permite crear una capa de abstracción entre el sistema operativo y el código fuente ejecutable de una aplicación determinada; de esta manera permite encapsular en una serie de librerías de clases a las API de Windows, llamadas a protocolos, librerías comunes, etc., a las mismas que se pueden acceder desde cualquier lenguaje soportado por esta plataforma de una manera transparente.

Como ventajas del uso de .NET se puede nombrar [15]:

- La disminución de los tiempos de desarrollo.
- La compatibilidad, portabilidad y reutilización de código entre plataformas operativas y lenguajes de desarrollo.
- La transparencia de ubicación de código.
- Un mejor control de versiones, tanto de la aplicación como de las librerías de clases.

.NET también proporciona un entorno de ejecución de aplicaciones; el cual proporciona una máquina virtual en el que las aplicaciones se ejecutan. En el mundo .NET, el entorno de ejecución se denomina CLR [10].

.NET Framework es un conjunto de APIs y una biblioteca compartida de código que los desarrolladores pueden llamar a la hora de desarrollar aplicaciones. Esto resulta extremadamente útil para los desarrolladores ya que evita escribir código desde cero. En palabras sencillas .NET contiene fragmentos de código que el desarrollador tiene que llamar cuando lo necesite; como ejemplo: en lugar de escribir el código para realizar ping de una computadora a otra que se encuentran en red, el desarrollador simplemente hará uso de la librería que realiza esta función.

“El marco de trabajo de .NET Framework se compone del CLR y la Base Class Library (BCL). Este marco de trabajo se superpone al Sistema Operativo y representa la única

interfaz existente de ejecución para las aplicaciones escritas para el entorno .NET [15]", como se puede observar en la Figura 1.5.

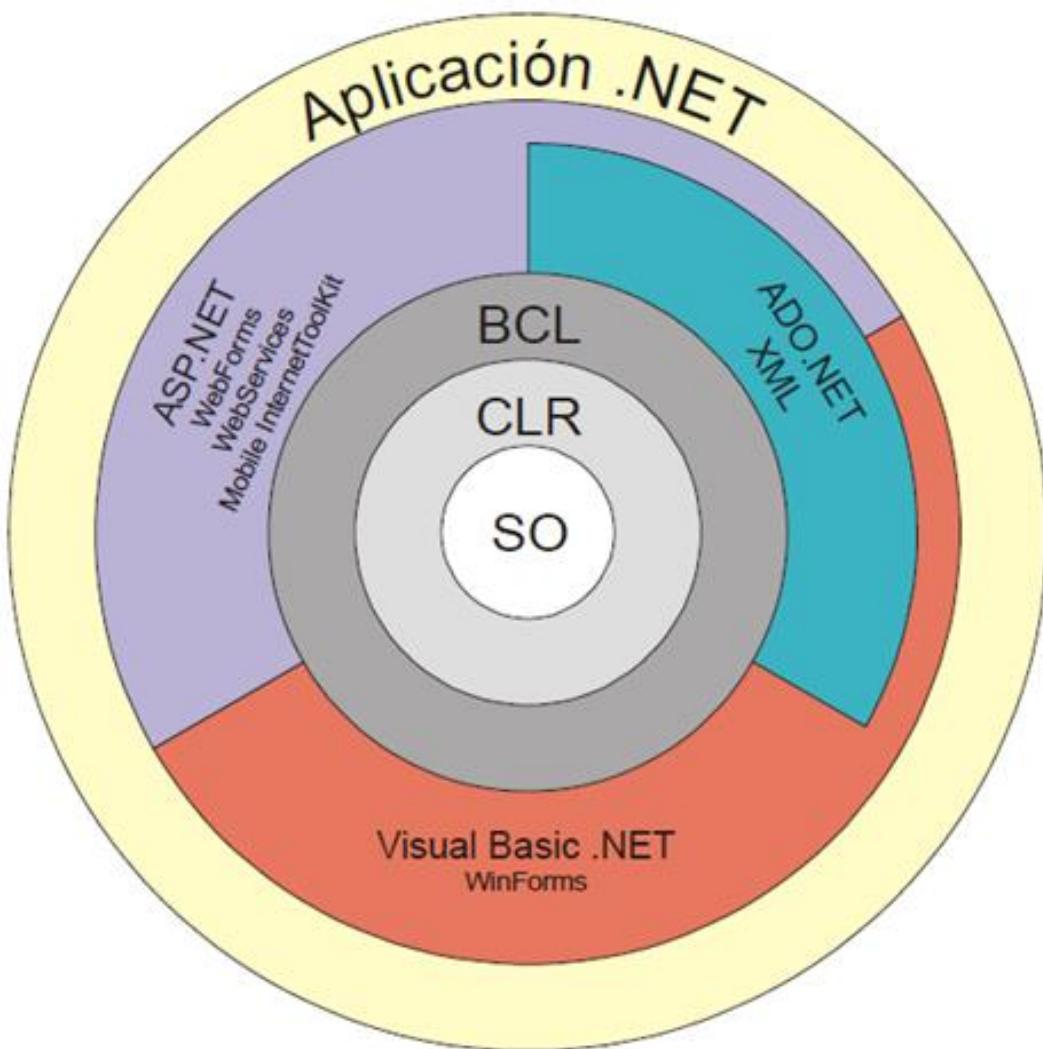


Figura 1.5. Comunicación entre los componentes de una aplicación .NET [15].

El CLR es la parte fundamental de .NET, debido a que se encarga de ejecutar las aplicaciones desarrolladas en esta plataforma. En la parte más baja se encuentra el sistema operativo, el cual anida al entorno de desarrollo; por encima se encuentra el cargador de clases, el cual permite cargar en la memoria las clases.

El MSIL se encarga de transformar el código fuente de los lenguajes soportados por .NET en un código intermedio para ser ejecutado, ayudándose del controlador de código y el recolector de basura para eliminar código innecesario y gestionar el tiempo de vida de los objetos y los recursos que utiliza una aplicación. En la Figura 1.6 se puede observar de una manera más detallada los componentes del CLR de una aplicación de .NET.



Figura 1.6. Entorno común de ejecución para lenguajes [32].

Por encima del CLR se encuentra el BCL; el BCL está formado por bibliotecas o APIs que pueden ser utilizadas por todos los lenguajes de programación soportados por .NET. Las bibliotecas que conforman el BCL pueden estar compuestas por varias clases y métodos, los mismos que proveen de funciones especiales a una aplicación, tales como: funciones matemáticas, funciones de correo, funciones de texto, etc.

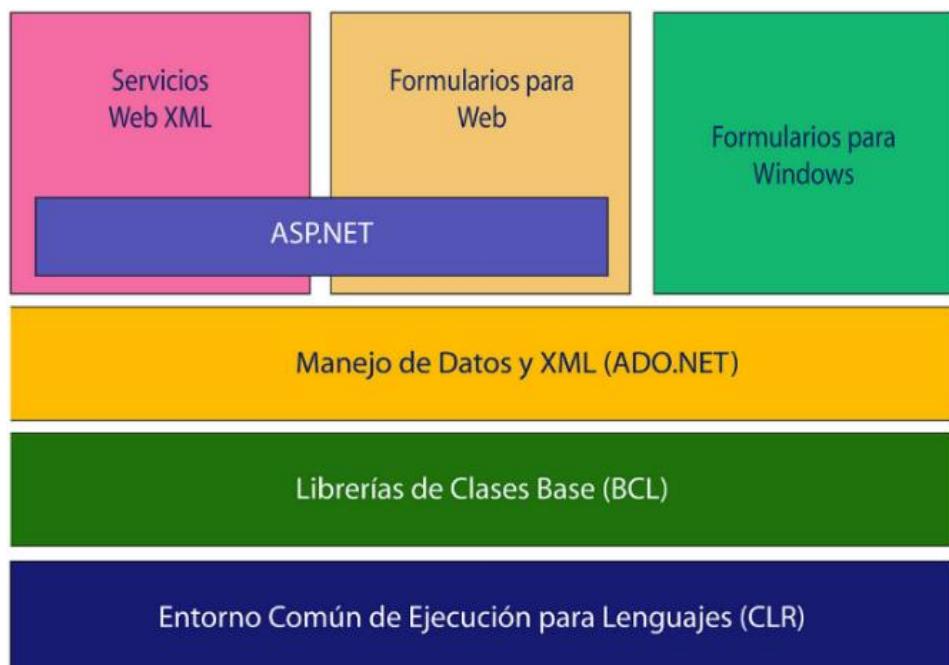


Figura 1.7. Biblioteca de clases base [32].

Adicionalmente, el BCL es el cual permite la interoperabilidad entre diferentes lenguajes, mediante el uso de funciones especializadas. En la Figura 1.7, se observa un esquema simplificado de los componentes del BCL; ya que en cada versión de .NET se van añadiendo, corrigiendo o eliminando funciones.

Para el caso de Visual Studio 2017, se detalla en la Tabla 1.1 los Frameworks con los que puede trabajar.

Tabla 1.1. Frameworks compatibles con Visual Studio 2017 [33].

FRAMEWORKS VISUAL STUDIO 2017
.NET Framework 2.0
.NET Framework 3.0
.NET Framework 3.5
.NET Framework 4.0
.NET Framework 4.5
.NET Framework 4.5.1
.NET Framework 4.5.2
.NET Framework 4.6
.NET Framework 4.6.1
.NET Framework 4.7
.NET Framework 4.7.1
.NET Framework 4.7.2

1.3.1.3. Windows Communication Foundation

WCF es un marco de Visual Studio para crear aplicaciones orientadas a servicios facilitando la creación de los puntos de conexión, “los puntos de conexión son los lugares donde los mensajes se envían o reciben (o ambos), y definen toda la información necesaria para el intercambio de mensajes [34]”; mediante WCF es posible crear aplicaciones Cliente – Servidor, donde uno de los extremos del servicio actúa como cliente enviando peticiones y el otro extremo actúa como servidor donde las peticiones son procesadas y retorna una respuesta al cliente; los mensajes que se intercambian pueden ser tan simples como un solo carácter, una palabra o incluso secuencia de datos binarios [35].

WCF incluye una serie de características, de las cuales las más importantes son:

- Orientación a servicios: WCF permite crear aplicaciones orientadas a servicios.

- Interoperabilidad: WCF implementa estándares de interoperabilidad del servicio Web.
- Varios patrones de mensajes: Los mensajes se pueden intercambiar de diferentes maneras. El más común es el de solicitud/respuesta. Existen otros tipos de mensajes, como el unidireccional, donde un extremo envía un mensaje sin esperar ninguna respuesta, el mensaje dúplex donde ambos puntos de conexión pueden enviar mensajes al otro de manera independiente.
- Seguridad: Es posible cifrar los mensajes para proteger la privacidad, así como obligar a los usuarios a que se autentiquen antes de permitirles recibir mensajes.
- Varios transportes y codificaciones: Los mensajes pueden enviarse con cualquiera de los protocolos y codificaciones integrados.
- Se admiten los canales TCP/IP de comunicación.
- Otros.

En la Figura 1.8 se observa las diferentes tecnologías combinadas para formar WCF; por ejemplo, la interoperabilidad con otras plataformas que posee WCF lo hace mediante ASMX, que son páginas de servicios web.

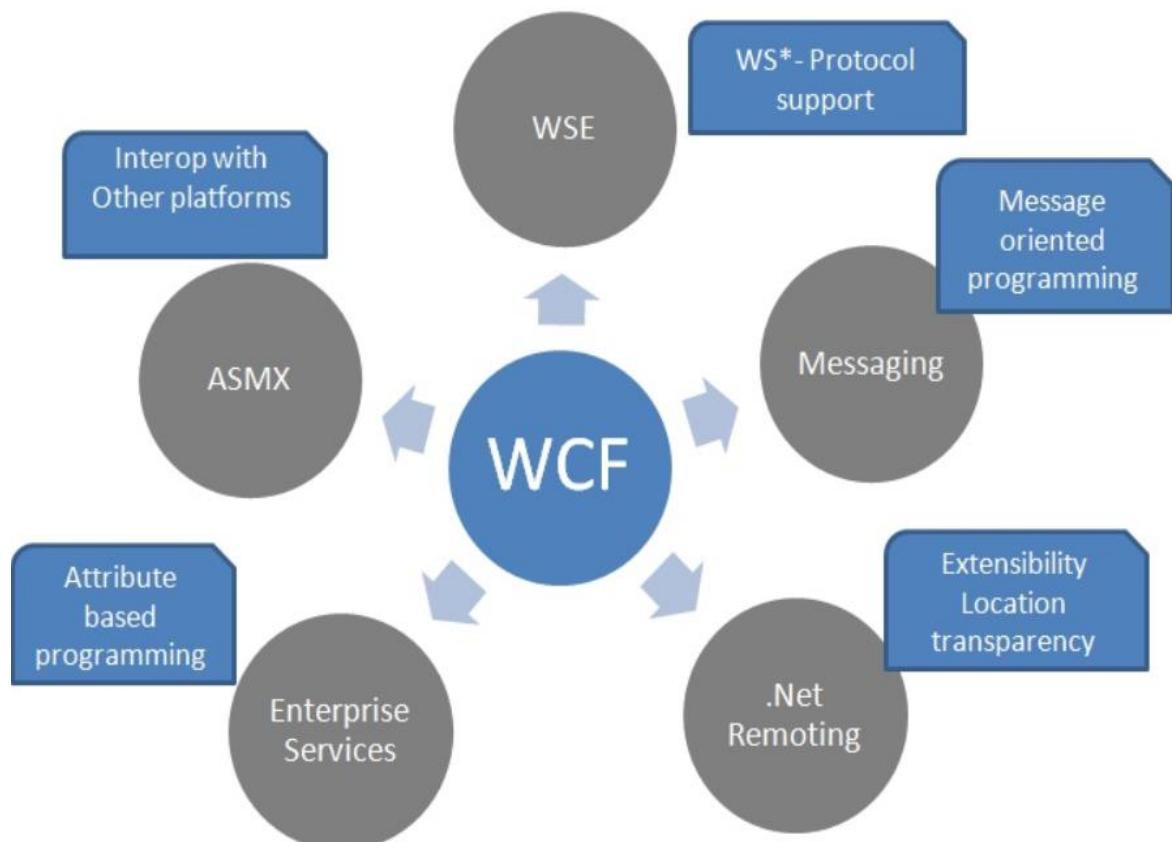


Figura 1.8. Tecnología combinada para formar WCF [36].

WCF es capaz de reconocer entre clientes, los cuales son aplicaciones que inician la comunicación y los servicios (servidores) que son las aplicaciones que esperan comunicarse con los clientes y responder a sus peticiones. Una misma aplicación puede actuar como cliente y servicio [34].

1.3.1.4. C Sharp

C Sharp (C#) es un lenguaje de programación orientado a objetos y tiene su origen en los lenguajes de la familia C; fue desarrollado por Microsoft en el año de 1999, por Andrés Hejlsberg, para permitir a los desarrolladores crear aplicaciones basadas en .NET Framework.

C# es un lenguaje altamente completo; se puede usar C# para crear aplicaciones cliente de Windows, componentes distribuidos, aplicaciones cliente-servidor, aplicaciones de base de datos y más cosas. Visual C# proporciona un editor de código avanzado, prácticos diseñadores de interfaz de usuario, un depurador integrado y otras herramientas que facilitan el desarrollo de aplicaciones basadas en el lenguaje C# y .NET Framework [37].

Algunas de las características de C#:

- Sintaxis sencilla.
- Espacio entre nombres.
- Integración con otros lenguajes.
- Multihilo.
- Orientado a Objetos.
- Otros.

En C# la sintaxis se simplifica mucho respecto a sus antecesores como C++; ya que C# es capaz de proporcionar nuevos tipos de valores, enumeraciones, delegados, expresiones lambda y acceso directo a la memoria que no se encuentra en otros lenguajes como en Java. Una de las fortalezas de C# es que permite el uso de LINQ lo que convierte una consulta fuertemente compleja en una construcción de lenguaje de primera clase; es decir, simplifica y hace más sencillas las consultas a una base de datos.

Con respecto a que C# es orientado a objetos; es importante mencionar, C# admite conceptos tales como: encapsulación, herencia y polimorfismo. Por ejemplo, una clase

puede heredar directamente de una clase primaria, pero puede implementar cualquier número de interfaces como se desee.

Las aplicaciones de C# en .NET Framework se compilan en un IL; los recursos IL, como mapas de bits y cadenas, se almacenan en disco en un archivo ejecutable denominado ensamblado, normalmente con la extensión .exe o .dll. Un ensamblado contiene un manifiesto que proporciona información sobre los tipos, la versión y los requisitos de seguridad del ensamblado [37]. La Figura 1.9 detalla el proceso de compilación y creación del archivo ejecutable.

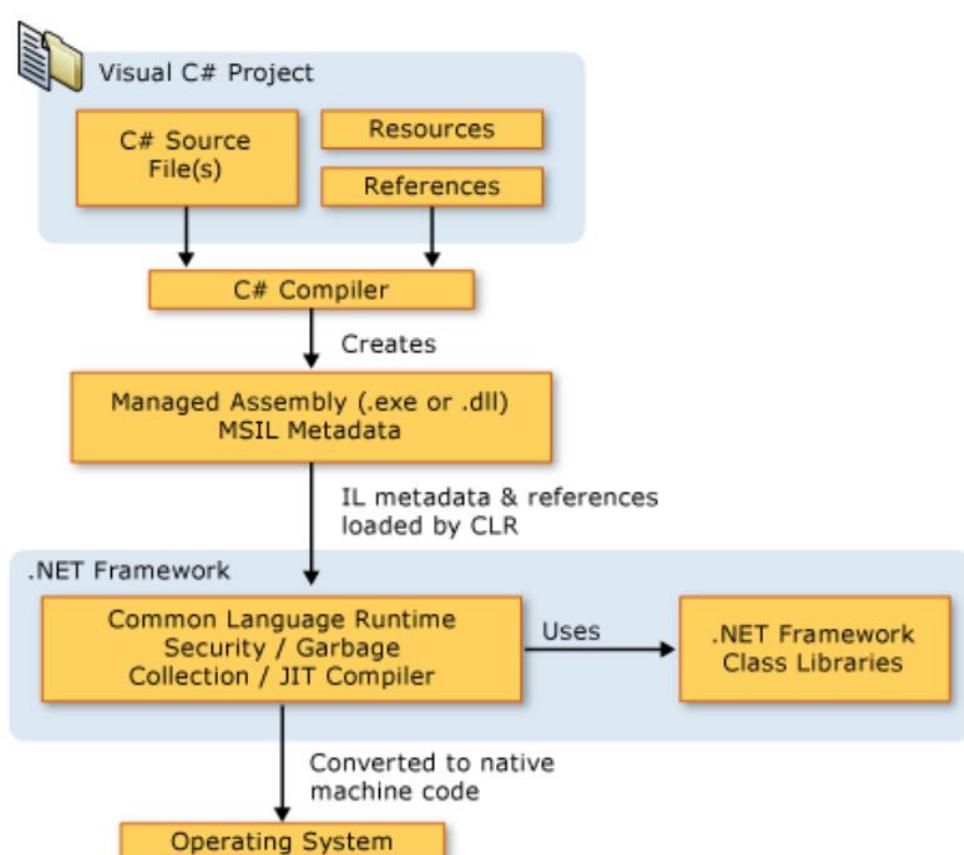


Figura 1.9. Compilación y creación del archivo ejecutable en C# [37].

Los tipos de datos, es algo muy importante a considerar, en .NET se utilizan los mismos tipos de datos para definir los valores de las variables y las constantes; aunque dependiendo del tipo de lenguaje pueden tener nombre diferente, así en C# se llama “int” e “Integer” en Visual Basic para llamar a un dato entero. Sin embargo, el CLR trata de la misma forma a ambas formas de llamar a un dato. Esto facilita el intercambio de información entre diferentes aplicaciones que son escritas en diferentes lenguajes pero que trabajan bajo .NET.

Los tipos de datos, básicos, están en la biblioteca de clases de .NET que se detallan en la Tabla 1.2:

Tabla 1.2. Tipos de datos, básicos, en la biblioteca de clases de .NET [38].

TIPO DE DATO	VB	C#	CONTIENE
Byte	Byte	Byte	Entero de 0 a 255
Int16	Short	Short	Entero de -32.768 a 32.767
Int32	Integer	Int	Entero de -2.147.483.648 a 2.147.483.647
Int64	Long	long	Entero de aproximadamente -9.2e18 a 9.2e18
Single	Single	float	Decimal de precisión simple de aproximadamente -3.48e38 a 3.48e38
Double	Double	double	Decimal de doble precisión de aproximadamente -1.8e308 a 1.8e308
Decimal	Decimal	decimal	Número de 128 bits fraccional de punto fijo que admite hasta 28 dígitos significativos
Char	Char	Char	Un único carácter Unicode de 16 bits
String	String	string	Una serie de caracteres de longitud variable de caracteres Unicode
Boolean	Boolean	bool	Un valor True o False
Object	Object	object	La clase base de todos los tipos .NET, puede contener cualquier tipo de datos u objetos (incluyendo los anteriores)

1.3.2. LINQ A SQL

LINQ a SQL es una extensión de ORM, el cual permite una conversión de los datos de un objeto a un formato correcto para poder realizar, por ejemplo, funciones básicas CRUD dentro de una base de datos. En la Figura 1.10, se muestra cómo se pueden realizar consultas a una base de datos mediante ORM.

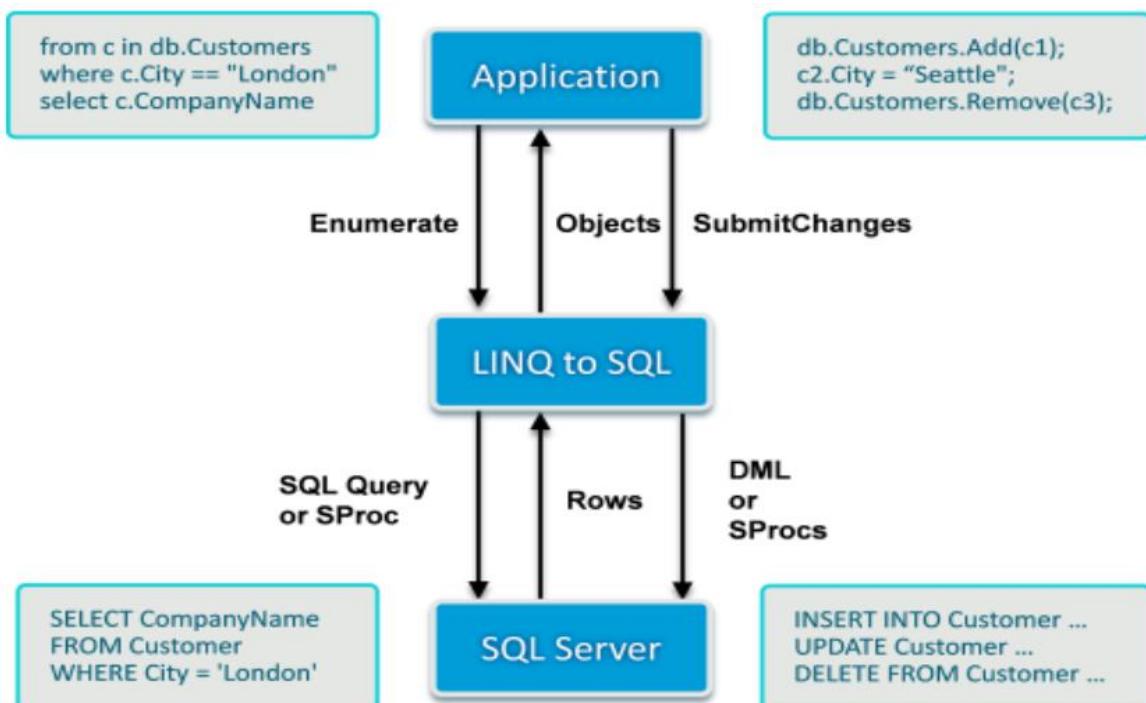


Figura 1.10. Mapeador de objetos relacionales [39].

LINQ a SQL consiste en mapear el esquema de una base de datos en clases, métodos y propiedades, tratando las tablas como colecciones de datos y las columnas como campos de las clases generadas, permitiendo definir consultas sobre los objetos [20].

1.3.2.1. LINQ

LINQ es un conjunto de herramientas diseñadas para reducir la complejidad de consultas SQL en Visual Studio y Microsoft .Net Framework 5, evitando así el uso de un lenguaje complicado y reduciendo el tiempo de consulta a la base de datos y los tiempos de proceso de la aplicación como tal.

Las consultas LINQ están introducidas dentro del código de la aplicación, lo que permite acceder a las bases de datos de SQL Server, archivos XML e incluso a base de datos de terceros alojadas en sitios distantes. LINQ funciona desde .NET Framework 3.0 lo que le permite trabajar con Visual Studio 2017 y a su vez con el lenguaje de programación orientada a objetos C#.

A continuación, se muestra una consulta donde se ingresa un nuevo usuario a una base de datos sin usar ORM, teniendo que escribir el código como se muestra en Código 1.1.

```

String query = "INSERT INTO clientes (id,nombre,email,pais)
command.Parameters.AddWithValue("@id","1")
command.Parameters.AddWithValue("@nombre", "nombre")
command.Parameters.AddWithValue("@email", "email")
command.Parameters.AddWithValue("@pais", "pais")
command.ExecuteNonQuery();

```

Código 1.1. Consulta SQL sin usar ORM a una base de datos [40].

Si se usa ORM, como se muestra en el Código 1.2 para realizar las consultas con LINQ, se puede visualizar que el código se reduce ampliamente y es más sencillo de implementarlo.

```

var cliente = new Cliente();
cliente.Id = "1";
cliente.Nombre = "nombre";
cliente.Email = "email";
cliente.Pais = "pais";
session.Save(customer);

```

Código 1.2. Consulta LINQ usando ORM a una base de datos [40].

LINQ permite una manera uniforme para el acceso y manipulación de los datos; por ejemplo, si se quiere realizar una consulta a una base de datos se usa SQL, para navegar en XML se usa DOM o XQuery, etc.; de esta manera LINQ facilita el uso de diferentes arquitecturas al unificar los procesos de consultas en uno solo. Se muestra en la Figura 1.11 como diferentes arquitecturas se integran para realizar consultas a base de datos mediante el uso de LINQ.

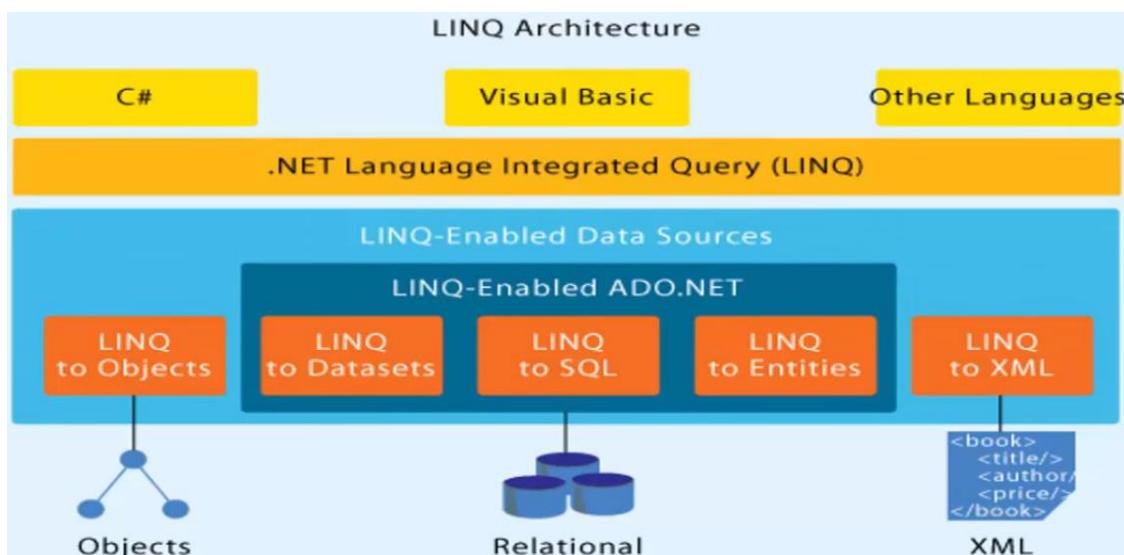


Figura 1.11. Arquitectura LINQ [3].

1.3.2.2. SQL

SQL es un lenguaje de consulta de base de datos normalizado de cuarta generación y utilizado por diferentes motores de base de datos para definir, gestionar y manipular los datos alojados en una base de datos.

SQL, el lenguaje relacional casi universalmente aplicado, es diferente de otros lenguajes computacionales como C, COBOL y Java, los cuales son de procedimiento. Un lenguaje de procedimiento define cómo las operaciones de una aplicación deben realizarse y el orden en el cual se realizan. Un lenguaje de no procedimiento, por otro lado, se refiere a los resultados de una operación; el entorno fundamental del software determina cómo se procesan las operaciones. Esto no quiere decir que SQL respalda a la funcionalidad de no procedimiento [14].

“Para utilizar SQL desde un lenguaje de programación se necesita sentencias especiales que permitan distinguir entre las instrucciones del lenguaje de programación y las sentencias de SQL [12]”, como es el caso de uso de LINQ en el lenguaje de programación de C# en Visual Studio.

En SQL se permite tres tipos de instrucciones:

- Lenguaje de definición de datos (DDL): las instrucciones se usan para crear, modificar o borrar objetos en una base de datos. Dentro de estas instrucciones están CREATE, DROP y ALTER.
- Lenguaje de control de datos (DCL): las instrucciones DCL permiten controlar quién o qué (usuario o aplicación) tiene acceso a objetos específicos en la base de datos. Dentro de estas instrucciones están: GRANT o REVOKE.
- Lenguaje de manipulación de datos (DML): las instrucciones DML se usan para recuperar, agregar, modificar o borrar datos almacenados en los objetos de una base de datos. Dentro de estas instrucciones están: SELECT, INSERT, UPDATE y DELETE.

1.3.3. OPENCV

Las siglas de OpenCV provienen de “Open Source Computer Vision Library”, lo que indica que es una librería destinada para el tratamiento de imágenes, ya sean almacenadas o en tiempo real.

“OpenCV (...) se publica bajo una licencia BSD y, por lo tanto, es gratis para uso académico y comercial. Tiene interfaces C++, Python y Java y es compatible con Windows, Linux, Mac OS, iOS y Android. OpenCV fue diseñado para la eficiencia computacional y con un fuerte enfoque en las aplicaciones en tiempo real. Escrita en C/C++ optimizado, la biblioteca puede aprovechar el procesamiento multi-core [41]”.

Dentro de las licencias de software libre y código abierto se encuentran, entre otras, con las licencias de GPL de GNU y la BSD; mientras licencias como la GPL de GNU permite usar, copiar y modificar el software garantizando a los usuarios que el software está protegido bajo la licencia o acuerdo de copyleft, lo que significa que otros usuarios que hagan uso del software no puedan apropiarse o cobrar tras hacer una distribución, ampliación o modificación del software.

La licencia BSD es una licencia permisiva, lo que significa que está en contraste con las licencias de copyleft; ya que BSD tiene menos restricciones que la GPL de GNU al permitir a los usuarios apropiarse y cobrar (o no hacerlo) [42] tras hacer una distribución, ampliación o modificación del software; en términos generales BSD permite usar código fuente de software libre y transformarlo en software no libre.

Uno de los objetivos de OpenCV es proporcionar una infraestructura de visión computarizada fácil de usar, lo cual ayuda a los usuarios a desarrollar aplicaciones de visión rápidamente [13]. Dentro de la biblioteca de OpenCV se puede encontrar más de 500 funciones que pueden ser usadas para la inspección de imágenes, seguridad, interfaces de usuario, calibración de cámaras, aprendizaje automático, reconocimiento de patrones, entre otras.

La visión computarizada que provee OpenCV, permite tomar una imagen de una cámara de video o una almacenada en el ordenador y transformarla en una nueva representación o modificación; una representación indica que se ha reconocido un objeto como puede ser uno o varios rostros en una imagen y una modificación se refiere al cambio parcial o total de una imagen como puede ser la modificación de una imagen a color a una en escala de grises.

Se puede apreciar lo antes mencionado en la Figura 1.12, donde la imagen de la izquierda se puede observar que se ha detectado cuatro rostros y en la imagen de la derecha se puede observar que el rostro central ha cambiado a escala de grises.

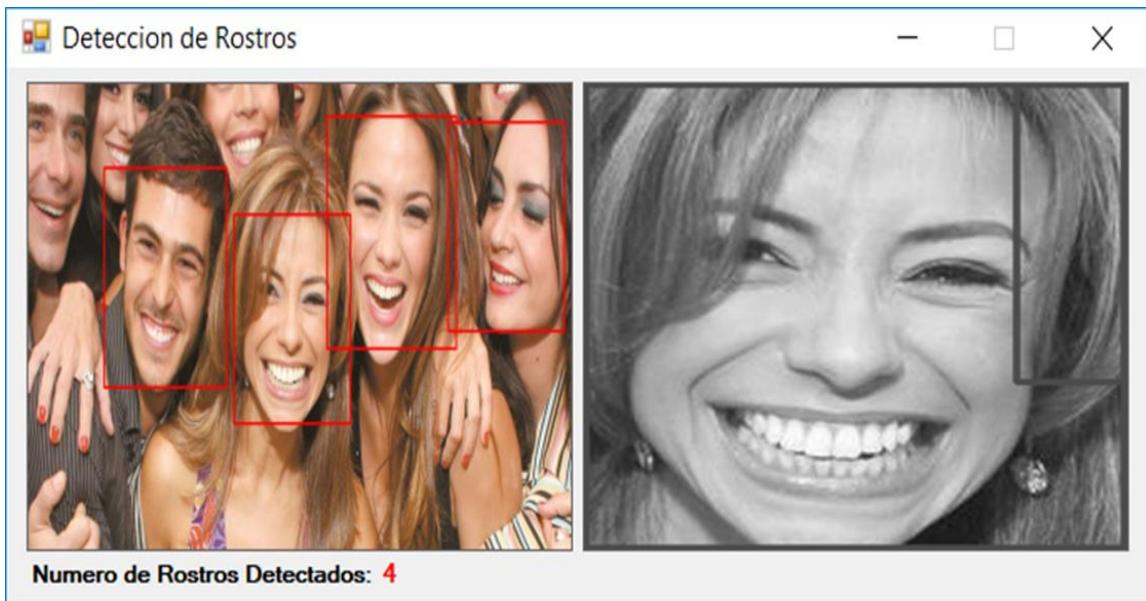


Figura 1.12. Representación y modificación de una imagen mediante el uso de OpenCV.

1.3.4. EMGUCV

EmguCV es un contenedor .Net multiplataforma para la biblioteca de procesamiento de imágenes OpenCV. Permitiendo que las funciones OpenCV sean llamadas desde lenguajes compatibles con .NET como C#, VB, C++, IronPython, etc. El envoltorio puede ser compilado por Visual Studio, Xamarin Studio y Unity, puede ejecutarse en Windows, Linux, Mac OS X, iOS, Android y Windows Phone [43].

Emgu CV está escrito completamente en C#. El beneficio es que puede compilarse en Mono y, por lo tanto, puede ejecutarse en cualquier plataforma compatible con Mono, incluyendo iOS, Android, Windows Phone, Mac OS X y Linux.

En la Tabla 1.3 se muestra las versiones de EmguCV y su compatibilidad con Visual Studio dentro de un ambiente Windows; se muestra también las características fundamentales y su tipo de licencia.

Otras ventajas del uso de EmguCV se muestran a continuación:

- Clase de imagen con color genérico y profundidad.
- Serialización de Imagen XML.
- La opción de usar la clase de imagen o las funciones de invocación directa de OpenCV.
- Operaciones genéricas en píxeles de imagen.

Tabla 1.3. Versiones de EmguCV [43].

Nombre	Emgu CV (Open Source)	Emgu CV para Windows (optimizado para uso comercial)	Emgu CV para Unity	
Herramientas de desarrollo	Visual Studio 2012 y superior	Visual Studio 2012 y superior	Visual Studio 2013 y superior	Unity 3D Pro v5.5 y superior
Plataforma	Windows	Windows	Aplicación UWP / Windows 8.1 Store	Windows Desktop Standalone
Arquitectura de CPU soportada	i386, x64	i386, x64	i386, x64	i386 (Editor y Standalone), x64 (Standalone)
Procesamiento de GPU CUDA	✓	✓	X	X
OpenCL (GPU & CPU)	✓	✓	X	✓
Tesseract OCR	✓	✓	✓	✓
Compilado con el compilador Intel C++, TBB e IPP	X	✓	X	X
Manejo de excepciones	✓	✓	✓	✓
Visualizador de depurador	✓	✓	X	X
Emgu.CV.UI	✓	✓	X	X
Licencia	GPL	Licencia comercial	Licencia comercial	

1.3.5. OZEKI CAMERA SDK

Ozeki Ltda. es una compañía de productos software y hardware que se estableció en el año 2000, en Debrecen – Hungría. Se especializa en facilitar la comunicación entre las personas y el acceso a dispositivos remotos. Entre uno de sus productos software, más sobresaliente, está el “Ozeki Camera SDK”.

Ozeki Camera SDK es un excelente kit de desarrollo de software para desarrolladores de C# .NET que le permite implementar aplicaciones de cámaras IP y cámaras WEB para sistemas de monitoreo en red, sistemas analíticos de video en red, detectores de movimiento, entre otros. Adicionalmente puede manejar cámaras USB y RTSP [44].

El SDK de la cámara OZEKI se basa en los estándares de ONVIF. “ONVIF es un foro líder y reconocido en la industria cuya misión es proporcionar y promover interfaces estandarizadas para la interoperabilidad efectiva de los productos de seguridad física basados en IP [45]”. Gracias a la flexibilidad de ONVIF ha hecho que esta tecnología se haga más popular. Los beneficios de ONVIF son:

- Interoperabilidad: permite la comunicación entre los productos de diferentes fabricantes.
- Flexibilidad: se puede trabajar con los productos de varios fabricantes.
- Calidad: el producto está constantemente actualizando y corrigiendo los errores que surgen.

Los requisitos para un desarrollador que desee crear una aplicación con el SDK de Ozeki son simples y se muestra en la Figura 1.13.



Figura 1.13. Requisitos básicos para crear una aplicación con Ozeki SDK [46].

Dentro de los requisitos cabe recalcar que Visual Studio debe tener una versión superior a la 2010 y el .NET Framework debe ser superior a la versión 4.0.

1.3.6. BIOMÉTRICOS

“La biometría es un método de reconocimiento de personas basado en sus características fisiológicas o de comportamiento. Se trata de un proceso similar al que habitualmente realiza el ser humano reconociendo e identificando a sus congéneres por su aspecto físico, su voz, su forma de andar, etc. [22]”.

Dentro del campo de los sistemas informáticos, la biometría informática es la aplicación de técnicas matemáticas sobre los individuos ya sea a sus características físicas (estáticas) o a su comportamiento (dinámicas) para lograr una autenticación; para ello se pueden usar diferentes métodos como son: el iris del ojo, las huellas dactilares, geometría de la mano, el rostro de la persona, etc., para el caso de características físicas y para el caso de comportamiento se puede usar el reconocimiento por voz o por firma, entre otros.

1.3.6.1. Etapas de un sistema biométrico

Como ya se ha nombrado, los métodos para la identificación de un individuo son muy diversos; pero estos métodos, en esencia, poseen las mismas etapas para el reconocimiento del individuo. Las etapas que se usan para la identificación se pueden reducir a dos etapas, como se muestra en la Figura 1.14.

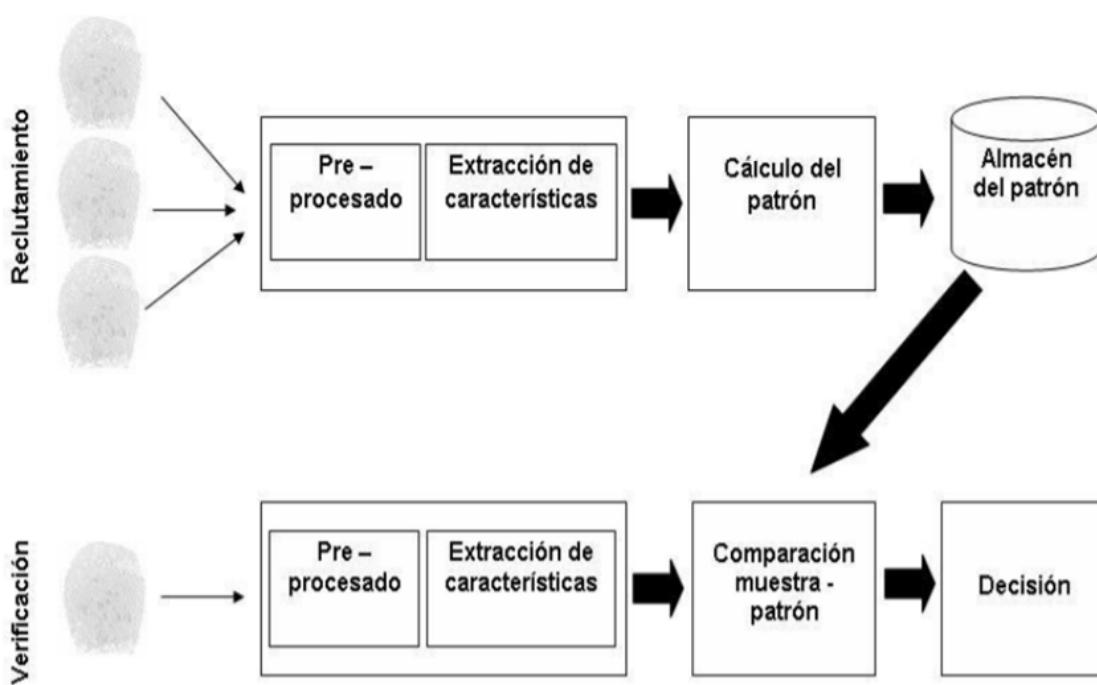


Figura 1.14. Etapas en un sistema de identificación biométrica [19].

En la Figura 1.14 se pueden observar dos etapas:

- Etapa de Reclutamiento o Inscripción: básicamente esta etapa es la encargada de tomar las muestras del individuo, las mismas que son procesadas y se obtiene un patrón único sobre el individuo; finalmente, este patrón es almacenado en una base de datos para una futura comparación. Si existen varias muestras aceptables del patrón se realiza un cálculo de la media del patrón, obteniendo así el patrón final a ser almacenado.
- Etapa de Verificación: al igual que en la etapa de inscripción, se toman muestras del individuo y son procesadas, obteniendo un patrón; este patrón finalmente, será comparado con uno almacenado en la base de datos, obteniendo un resultado positivo o negativo.

Cada etapa está conformada por varias sub-etapas las mismas que tienen un propósito en específico, estas son:

- Captura (En la imagen representado con la huella dactilar): aunque no se muestra directamente en la Figura 1.14, esta hace referencia al elemento tecnológico para obtener las características del individuo; es decir, este elemento corresponde a una cámara de video, un lector dactilar, etc.
- Pre-procesado: esta sub-etapa se encarga de adecuar la información obtenida para facilitar su tratamiento posterior. Por ejemplo, aquí es donde se delimitan los bordes de una imagen, se la amplia o reduce, etc.
- Extracción de características: la sub-etapa más importante; ya que aquí es donde son extraídas las características más significativas, el patrón, del individuo. Este patrón es almacenado en forma de plantilla en una base de datos. Es en esta sub-etapa donde se fundamenta la capacidad del sistema biométrico en reconocer entre individuos.
- Comparación: se realizan las tres sub-etapas anteriormente mencionadas obteniendo un patrón del individuo; este patrón es comparado con el almacenado en la base de datos. Esta comparación no se trata de una comparación binaria o de igualdad; más bien se trata de una comparación de variaciones o diferencias entre el patrón obtenido y el almacenado. Así para determinar una decisión positiva o negativa se debe crear un umbral de aceptación, esta aceptación está basada en la Teoría de Reconocimiento de Patrones: Distancia Euclídea, Distancia de Mahalanobis, Distancia de Hamming, Estadísticas utilizando funciones de distribución, clasificadores bayesianos, técnicas basadas en modelado de problemas como Redes Neuronales y Modelos de Mezclas de Gaussianas.

Para la etapa de verificación cabe aclarar que existen dos procesos diferentes para hacerlo, estos son: Identificación y Autenticación. La identificación es tomar el patrón obtenido y hacer una comparación con una serie de patrones almacenados en la base de datos, es decir una comparación uno a varios, donde el resultado de las distintas comparaciones resulta en la identificación del individuo; mientras que la autenticación es donde se toma el patrón obtenido y se lo compara con uno ya guardado en la base de datos, es decir se hace una comparación uno a uno. Esto se muestra en la Figura 1.15.

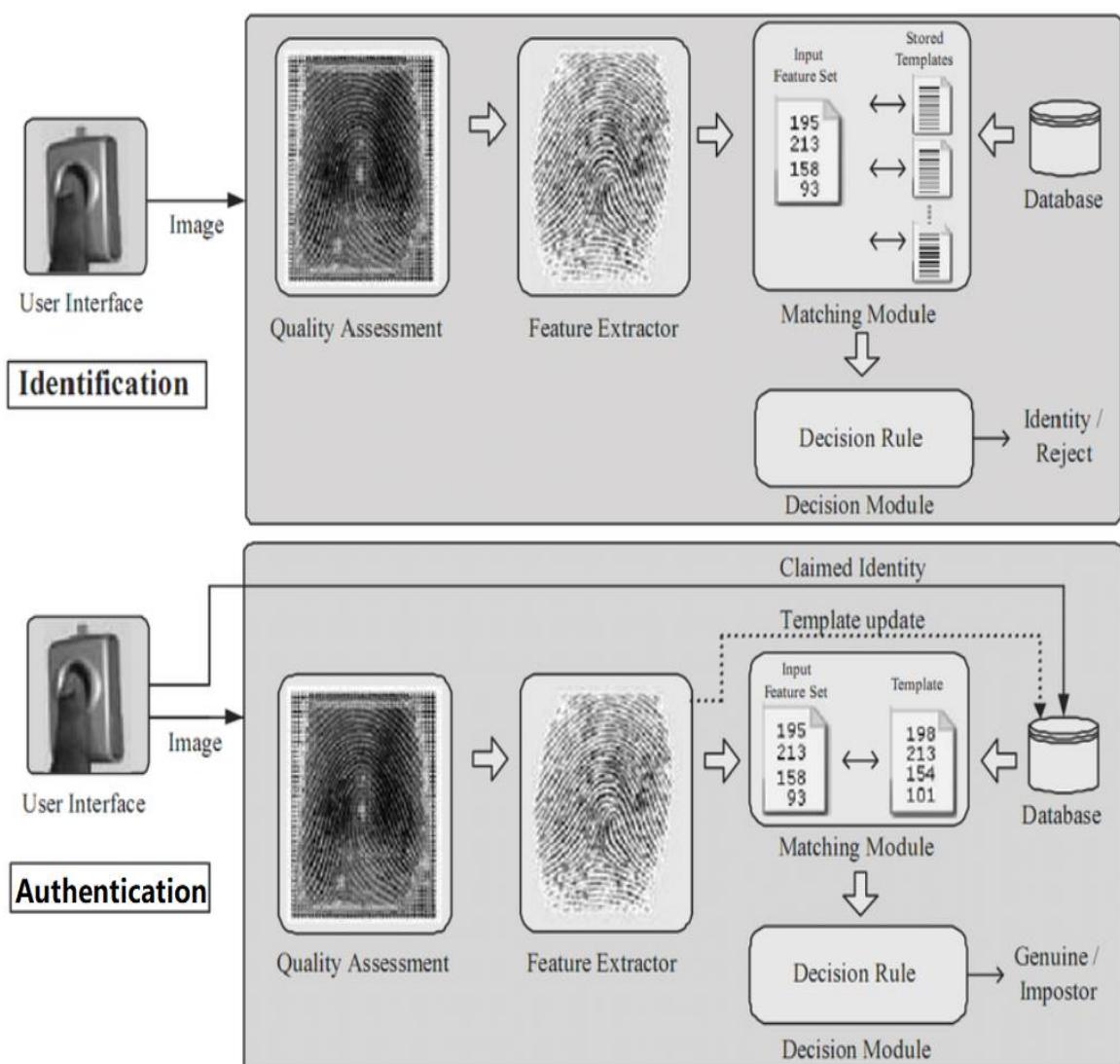


Figura 1.15. Identificación (arriba) y autenticación (abajo), diferencias entre las fases [19].

El proceso de identificación es más rápido que el proceso de autenticación; pero significa que en primera instancia se debe conocer presuntamente la identidad del individuo que hace uso del sistema biométrico; mientras que, para la autenticación, que es más lenta, resulta en no conocer la presunta identidad del individuo.

1.3.6.2. Rendimiento de un sistema biométrico

Como se nombró anteriormente, todo sistema biométrico tiene un índice de aceptación o umbral dictado por la Teoría de Reconocimiento de Patrones; esto indica nada más y nada menos el rendimiento que tiene el sistema biométrico.

Si el umbral del sistema biométrico es demasiado bajo, existe la posibilidad de que personas no autorizadas puedan ingresar al sitio custodiado por el sistema biométrico y si por el contrario el umbral es demasiado alto puede suscitarse de que los usuarios autorizados del sistema no puedan ingresar; por lo cual es necesario establecer un umbral adecuado al método escogido del sistema biométrico. Los sistemas biométricos actuales tienen un rendimiento que va desde el 60% para los más bajos hasta el 99.9% para los más exigentes [47].

El rendimiento de un sistema biométrico está dado en los siguientes términos:

- Tasa de Falsa Aceptación (FAR): indica cual es la tasa de que personas no autorizadas puedan ingresar.
- Tasa de Falso Rechazo (FRR): también conocida como FNMR, indica la tasa de rechazo a persona autorizada.
- Tasa de Error de Reclutamiento (FTE): también conocida como FER, indica el porcentaje de usuarios que no pueden registrarse en el sistema, esto es debido al método biométrico utilizado (estático o dinámico); ya que según el método se deberían tomar varias muestras del patrón del individuo, este es el caso para sistemas basados en voz, donde se debe considerar la pronunciación que realiza el individuo.
- Tasa de Error Igual (EER): indica la tasa donde el FAR y el FRR son iguales.
- Tasa de Error de Cruce (CER): indica la sensibilidad del sistema biométrico, es el punto donde se cruza el FAR y el FRR.

Los valores FAR y FRR pueden ser observados de manera más detallada en la Figura 1.16, donde se detalla una gráfica de compensación de error (DET). La curva de rendimiento es calculada usando los resultados después de la comparación con la base de datos, los datos son mostrados en una escala de desviación normal. En la Figura 1.17, se muestra el gráfico donde la Tasa de Falsa Aceptación y la Tasa de Falso Rechazo se cruzan; esto indica la sensibilidad del sistema biométrico y por ende indica el índice de error que el mismo posee.

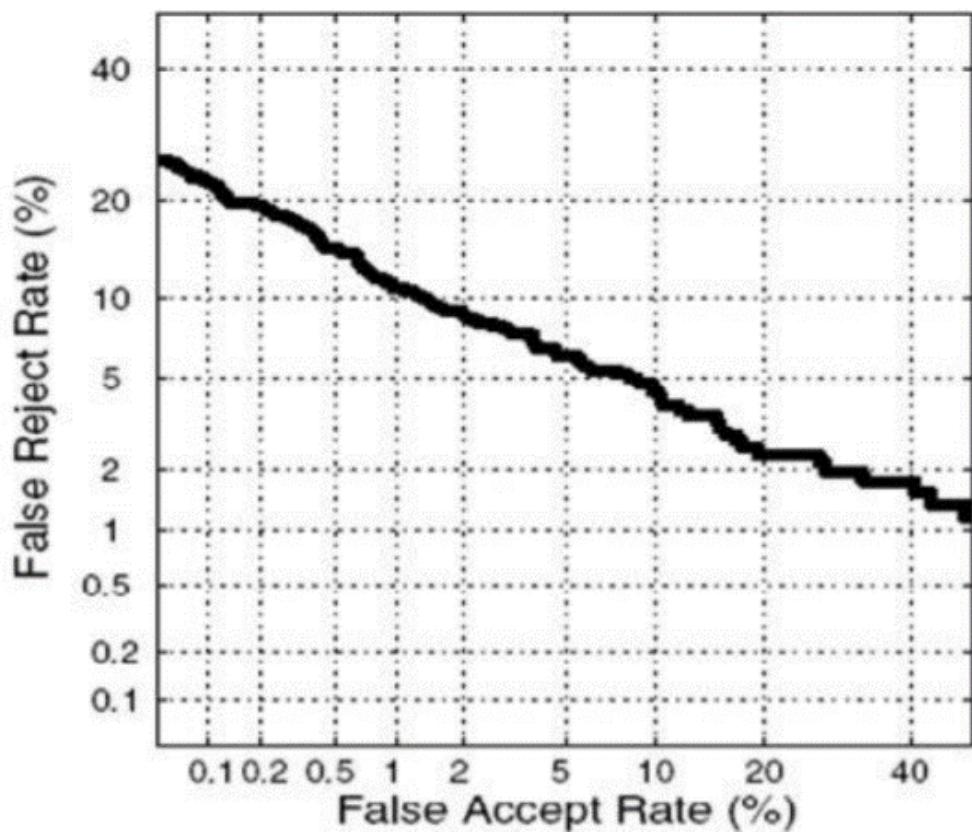


Figura 1.16. Rendimiento de un sistema biométrico resumido en un gráfico DET [19].

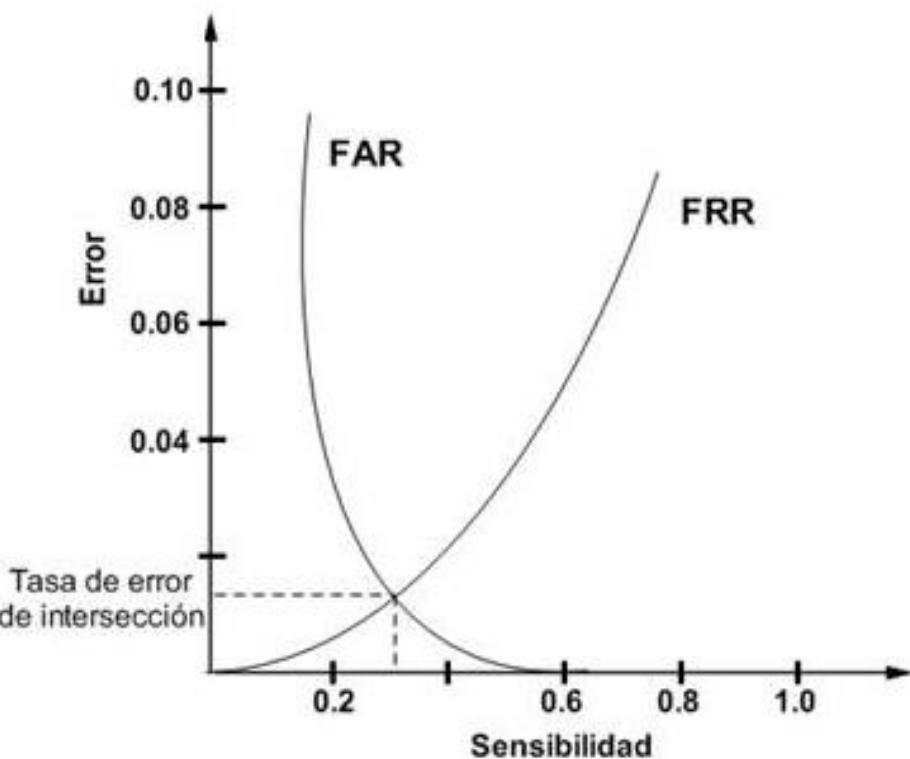


Figura 1.17. Sensibilidad del sistema biométrico.

1.3.6.3. Requisitos de un sistema biométrico

Dependiendo de la tecnología biométrica utilizada, estática o dinámica, se extrae un patrón único para cada persona como la voz o patrón de la cara, para posteriores comparaciones con un patrón previamente almacenado en una base de datos. Para garantizar que el patrón obtenido sea único, el sistema biométrico debe cumplir con los siguientes requisitos:

- Universalidad: todos los individuos deben poseer la misma característica a medir.
- Univocidad: la característica obtenida del individuo lo debe distinguir.
- Permanencia: la característica del individuo debe ser la misma en cualquier momento y lugar.
- Cuantificación: la característica del individuo debe poder medirse.

Para el caso del hardware y software que utilice el sistema biométrico debe cumplir con las siguientes características [22]:

- Rendimiento: debe tener un buen nivel de exactitud.
- Aceptación: el usuario debe estar de acuerdo con la fiabilidad del sistema.
- Seguridad: el sistema debe ser resistente a la usurpación.

1.3.6.4. Ventajas y desventajas de los sistemas biométricos

Los sistemas biométricos no son perfectos, aunque están cerca, por lo cual proporcionan ciertas ventajas y desventajas al usuario final que ocupe un determinado sistema biométrico. Se nombra las ventajas y desventajas más notables a continuación.

Ventajas:

- No es necesario andar a cargar tarjetas o credenciales para el ingreso a un determinado sitio.
- Las tecnologías de acceso mediante sistemas biométricos son más seguras que las habituales como el usuario y contraseña, las cuales deben ser recordados por el individuo y de llegar a olvidarse se deben pasar por ciertos protocolos para poder recuperarlos.
- Es más barato el mantenimiento; ya que solo se debe dar mantenimiento al lector biométrico.
- Las características físicas de una persona no pueden ser trasmítidas a otra persona.

Desventajas:

- Robo de identidad: dado el caso que el sistema biométrico sea vulnerado, los patrones biométricos no pueden ser cambiados. Por ejemplo: si una cuenta de correo es vulnerada, el sistema provee la opción de cambiar la contraseña; no es así en los sistemas biométricos ya que el patrón biométrico es único y no se lo puede modificar.
- Privacidad: al tener un patrón único, esto provee a entidades gubernamentales un fácil rastreo de un individuo y que posiblemente los datos privados sean vigilados u observados.

En la Tabla 1.4, se puede observar las ventajas y desventajas de un sistema biométrico, según sea el método escogido. No se consideran todos los métodos, solo los más usados.

Tabla 1.4. Comparación entre algunos métodos biométricos [48].

	Ojo (Iris)	Huellas dactilares	Geometría de la mano	Escritura y firma	Voz	Cara 2D
Fiabilidad	Muy alta	Muy Alta	Alta	Media	Alta	Media
Facilidad de uso	Media	Alta	Alta	Alta	Alta	Alta
Prevención de ataques	Muy alta	Alta	Alta	Media	Media	Media
Aceptación	Media	Alta	Alta	Muy Alta	Alta	Muy alta
Estabilidad	Alta	Alta	Media	Baja	Media	Media

1.3.7. DETECCIÓN Y RECONOCIMIENTO FACIAL

Se debe distinguir, con precisión, lo que es Detección Facial de lo que es Reconocimiento Facial ya que el reconocimiento facial se basa en la detección facial. Para un sistema biométrico que se base en esta tecnología primero se debe detectar el rostro de una persona y luego hacer el reconocimiento del rostro detectado en una base de datos.

1.3.7.1. Detección facial

La detección facial es una tecnología que permite determinar el lugar y el tamaño de rostros humanos en imágenes o videos; esto viene a ser un caso particular de la detección de objetos. Si bien para un ser humano, realizar la tarea de ubicar un objeto es relativamente sencillo, no lo es así para una computadora; ya que esta depende de varios factores como lo son: la posición del rostro, iluminación, expresiones faciales, occlusiones (lentes, gorros, parches, etc.), detalles faciales (bigote, cicatrices, etc.).

Para un sistema biométrico basado en reconocimiento facial, el primer paso es la detección del rostro; esto no resulta sencillo para un computador ya que debe identificar el objeto requerido sin que los factores antes mencionados lo afecten. Para ello hace uso de varios métodos que le permiten hacer la detección del objeto, los métodos más usados son [4]:

- Métodos basados en conocimiento: este método permite medir las distancias entre las partes que componen un objeto. Como ejemplo, se puede mencionar al ser humano al cual se le puede medir la distancia entre los ojos, nariz y labios.
- Métodos basados en características invariantes: este método toma en cuenta los detalles de un objeto que no varían, independiente de la posición de la cámara de video. Por ejemplo, se puede nombrar la textura o color de la piel de un individuo.
- Métodos basados en moldes o patrones: es la comparación entre un patrón de ingreso y uno almacenado previamente, cuyo objetivo es capturar las características del rostro; es decir, se da un modelado geométrico.

El algoritmo de Viola – Jones, es el algoritmo para detectar rostros con mayor índice de acierto [17], con un 99.9% de acierto; este algoritmo es muy utilizado ya que no requiere de altos procesos computacionales. Está constituido por tres partes: el primero es la imagen integral, el segundo es el clasificador en cascada, que garantiza una discriminación rápida al descartar objetos de fondo que no son de interés para el análisis; y el tercero es el entrenador de clasificadores, que está basado en Adaboost, el cual es un meta-algoritmo de aprendizaje automático, que garantiza seleccionar las características más importantes de todo el conjunto [6]. Gracias a su potencia y velocidad este algoritmo es implementado en la librería de OpenCV.

El procesamiento que realiza lo hace en imágenes en escala de grises, la misma que es transformada en una imagen integral, lo que significa que se realiza una representación intermedia de la imagen mediante una Tabla de Área Sumada (Summed Area Table) y a

su vez permite un cómputo más rápido. En la Figura 1.18, se puede observar la imagen original en representación de grises numérico y a la derecha se puede observar la imagen integral.

0.1	0.1	0.2	0.1	0.7	0.1
0.2	0.3	0.2	0.7	0.8	0.2
0.1	0.4	0.3	0.3	0.1	0.3
0.1	0.5	0.1	0.1	0.2	0.8
0.1	0.4	0.8	0.5	0.6	0.5

→

0.1	0.2	0.4	0.5	1.2	1.3
0.3	0.7	1.1	1.9	3.4	3.7
0.4	1.2	1.9	3.0	4.6	5.2
0.5	1.7	2.5	3.7	5.3	6.7
0.6	2.3	3.9	5.6	8.0	9.9

Figura 1.18. Cambio de imagen original a una imagen integral.

La imagen integral es una matriz de igual tamaño que la original, esta matriz resulta de la suma de los elementos, pixeles en una imagen, arriba y a la izquierda del punto “X” seleccionado, como se puede observar en la Ecuación 1.1 (Integral de la imagen) [17] y en la Ecuación 2.2 (Tabla de área sumada) [17] se detalla las fórmulas que las describen en forma recursiva.

$$II(x, y) = \sum_{x' \leq x, y' \leq y} I(x', y') \quad (2.1)$$

La integral de la imagen, en forma recursiva, es representada por:

$$S(x, y) = S(x, y - 1) + I(x, y) \quad (2.2)$$

$$II(x, y) = II(x - 1, y) + S(x, y)$$

Dónde:

- $II(x', y')$ Es la integral de la imagen.
- $I(x', y')$ Es la imagen original.
- $S(x, y)$ Es la suma acumulada en fila.
- $S(x, -1) = 0$ y $II(-1, y)=0$.

El clasificador en cascada (haar), es un conjunto de descriptores que permiten obtener información de un área en particular mediante operaciones matemáticas, esto se lo realiza aplicando filtros haar en el algoritmo de Viola – Jones, estos filtros pueden ser de tres tipos, como se puede observar en la Figura 1.19:

- Característica de dos rectángulos: es la diferencia entre la suma de los píxeles de ambas regiones rectangulares.
- Característica de tres rectángulos: Es la suma de los píxeles de los rectángulos exteriores menos la suma del rectángulo central.
- Característica de cuatro rectángulos: Es la diferencia entre los pares diagonales de los rectángulos.

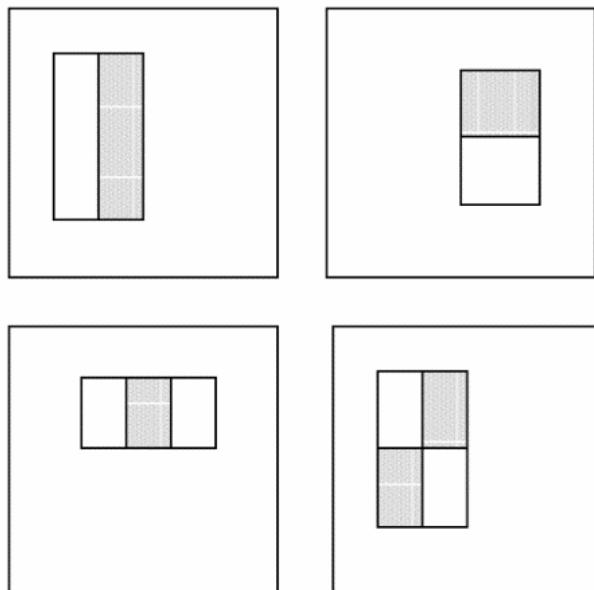


Figura 1.19. Características haar [17].

Para determinar si en una imagen se encuentra un rostro o no, el algoritmo de Viola – Jones en conjunto con Adaboost divide la imagen integral en subregiones de diferentes tamaños, donde se utiliza un clasificador en cascada determinando si en cada subregión existe o no un rostro. Este algoritmo funciona mejor con rostros frontales; ya que en rostros con vista de perfil aportan variaciones a la plantilla que no puede manejar.

Las características elegidas por AdaBoost son significativas y de fácil interpretación. La elección de la primera característica se basa en la propiedad que la región de los ojos es más oscura que la región de las mejillas. La segunda característica se basa en que los ojos son más oscuros que la zona de la nariz [4]. Se puede observar lo mencionado en la Figura 1.20. Ver el “Anexo B” para más detalle del funcionamiento del algoritmo.



Figura 1.20. Las dos características de haar más significativas seleccionadas por AdaBoost [17].

1.3.7.2. Reconocimiento facial

Un sistema biométrico basado en reconocimiento facial, es una aplicación dirigida por un ordenador, que se encarga de identificar a una persona automáticamente de una imagen digital; esta imagen puede ser obtenida por una cámara de video en tiempo real o una que esté almacenada en el ordenador. El ordenador se encarga de hacer una comparación entre la imagen obtenida y una almacenada en una base de datos [16].

Para lograr el reconocimiento facial de un individuo, es necesario de cuatro pasos:

- Detección facial: mediante el uso de un algoritmo, como el de Viola – Jones, se procede a detectar la existencia de un rostro en una imagen o video.
- Alineación facial: localiza los rasgos del rostro en la imagen; esto significa que se procede a ubicar los parámetros característicos del rostro como la boca, la nariz o ceja, así como las distancias entre estas. Aquí también se elige el tamaño de la imagen, así como la gama de colores.
- Extracción de las características faciales: proporciona la información necesaria del rostro detectado para distinguirlo entre personas diferentes.
- Reconocimiento: el patrón extraído en los pasos anteriores se procede con la comparación con uno almacenado en una base de datos; para ello se usa un algoritmo de reconocimiento facial.

En la Figura 1.21, se puede observar a detalle el procedimiento realizado para realizar el reconocimiento facial de una persona.

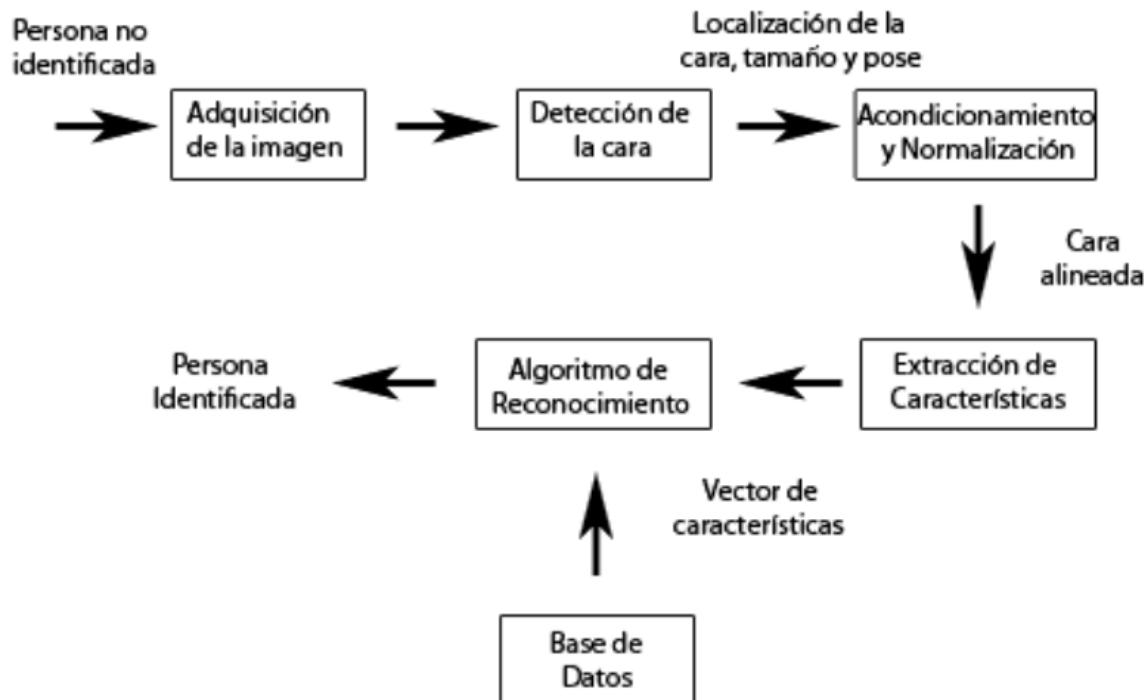


Figura 1.21. Procedimiento para el reconocimiento facial [23].

Un aspecto importante para el reconocimiento facial es el poder distinguir entre el fondo de la imagen y el rostro en la imagen; para ello se debe reconocer los “puntos duros” del rostro y tratarlos como puntos fijos a los cuales se procede a medir la distancia entre ellos y finalmente comparar estos puntos o patrón con el almacenado en la base de datos. Existen unos 80 puntos duros en un rostro de los cuales los más significativos son los puntos de las cejas, ojos, nariz y boca; en la Figura 1.22 se puede observar estos puntos.



Figura 1.22. Puntos duros en un rostro [22].

Para lograr la adquisición de las imágenes, generalmente, se hace uso de un sensor CCD que por defecto ya vienen incluidos en las cámaras de video digitales. La imagen capturada es una matriz bidimensional en la cual cada elemento es un pixel. El Dispositivo de Carga Acoplada – CCD, es un circuito integrado que contiene un número determinado de condensadores enlazados o acoplados, llamadas celdas fotoeléctricas; estas celdas ayudan a registrar la imagen. La capacidad de resolución o detalle de la imagen depende del número de células fotoeléctricas, este número se expresa en píxeles como se muestra en la Figura 1.23.



Figura 1.23. Diferentes pixelados en una imagen [11].

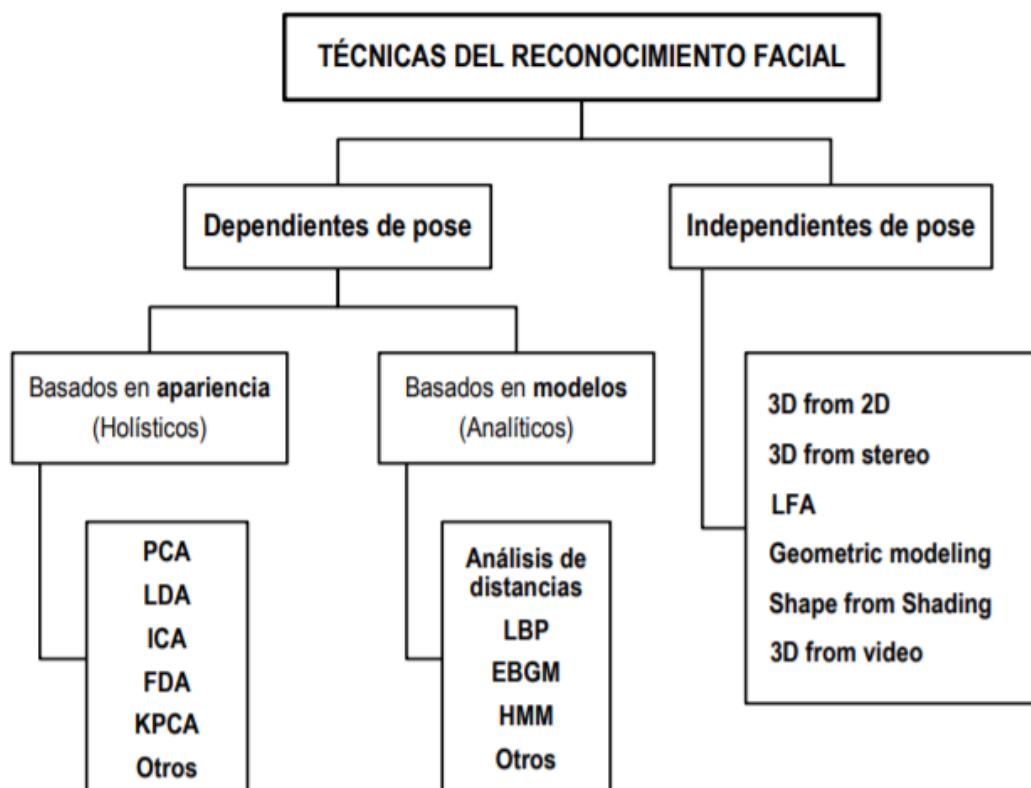


Figura 1.24. Técnicas del reconocimiento facial [8].

Existen varios tipos de algoritmo para el reconocimiento facial, como se puede ver en la Figura 1.24. “En primer lugar, las técnicas basadas en apariencia, que analizan la textura de la imagen a partir de la cual se aplicarán diferentes técnicas estadísticas y se extraerá la información. En segundo lugar, las técnicas basadas en modelos, que extraen las características tanto de la forma del rostro como de la textura [8]”. Y en tercer lugar las técnicas independientes de la pose la cuales realizan modelados en 3D y que requieren de equipos sofisticados para el reconocimiento facial.

La librería OpenCV cuenta con clases (`Facerecognizer`) que facilitan el trabajo a la hora de hacer reconocimiento facial, la clase `facerecognizer` implementa tres algoritmos: `EigenFaces`, `FisherFaces` y `LBHP`, se puede implementar el que mejor se ajuste a nuestras necesidades.

El algoritmo `EigenFaces`, es un algoritmo que toma en consideración las características comunes de las personas como son: la nariz, boca, cejas, ojos y las distancias entre ellos, estos componentes en común son llamados `EigenFaces`.

El algoritmo `EigenFaces` trabaja de la siguiente manera:

- Imágenes de Entrenamiento: se debe poseer imágenes de entrenamiento, es decir una colección de imágenes de diferentes personas a las cuales se quiere realizar el reconocimiento facial. Estas imágenes deben poseer diferentes posturas de la persona y diferentes condiciones de iluminación. Se puede observar esto en la Figura 1.25.
- Operaciones Algebraicas: el algoritmo `EigenFaces` no trabaja directamente con las imágenes, sino que las transforma en una matriz de $n \times n$ pixeles, donde cada pixel tiene un valor entre 0 y 255. A esta matriz se le aplican una serie de operaciones algebraicas obteniendo así un rostro nuevo, como se puede observar en la Figura 1.26; esta nueva imagen puede ser usada para compararla con rostros existentes.

El algoritmo `FisherFaces` tiene en cuenta cómo se refleja la luz y las expresiones faciales en el rostro de una persona. `FisherFaces` clasifica y reduce la dimensión de las caras utilizando el método Discriminante Lineal de Fisher (FLD) y PCA (conocido como `EigenFaces`). Este método crea una proyección lineal que maximiza las diferentes imágenes de caras proyectadas; pudiendo así hacer comparaciones posteriores mediante el uso de la distancia euclíadiana. Se observa en detalle en la Figura 1.27.



Figura 1.25. Imágenes de entrenamiento [4].



Figura 1.26. Conjunto de EigenFaces [4].



Figura 1.27. Cuadro Fisher de una imagen de entrada [4].

El algoritmo LBHP se basa en el algoritmo LBP; ya que se ha determinado que cuando el LBP se combina con los histogramas del descriptor de gradientes, mejora considerablemente el rendimiento de detección. LBP es un operador de textura simple pero muy eficiente que etiqueta los píxeles de una imagen mediante el umbral de la vecindad de cada píxel, creando una imagen binaria en escala de grises.

Usando el LBP combinado con histogramas se puede representar las imágenes de la cara con un simple vector de datos, lo cual facilita el reconocimiento de rostros. LBPH usa 4 parámetros [49]:

- Radio: el radio se utiliza para construir el patrón binario local circular y representa el radio alrededor del píxel central. Por lo general, se establece en 1.
- Vecinos: el número de puntos de muestra para construir el patrón binario local circular. Tenga en cuenta que cuantos más puntos de muestra incluya, mayor será el costo computacional. Por lo general, se establece en 8.
- Cuadrícula X: el número de celdas en la dirección horizontal. Cuantas más celdas, más fina sea la cuadrícula, mayor será la dimensionalidad del vector de características resultante. Por lo general, se establece en 8.

- Cuadrícula Y: el número de celdas en la dirección vertical. Cuantas más celdas, más fina sea la cuadrícula, mayor será la dimensionalidad del vector de características resultante. Por lo general, se establece en 8.

Al igual que EigenFaces, LBP necesita de una serie de imágenes conocidas como Imágenes de Entrenamiento, con las cuales el algoritmo podrá reconocer una imagen de entrada y darle una salida. Luego de tener las Imágenes de Entrenamiento, se debe aplicar el algoritmo LBP a cada imagen; finalmente, se debe obtener la imagen con la cual se desea comparar; esta imagen debe estar en escala de grises o ser transformada como tal.

Con la imagen a comparar; primero se toma un pixel y sus vecinos creando así una matriz de 3x3, y representando cada pixel según su intensidad con un valor entre 0 y 255. Luego se toma el valor central de la matriz, el cual será usado como umbral; este valor será usado para definir los 8 valores vecinos. Si la intensidad del píxel central es mayor o igual que su vecino, entonces se establece el valor en 0; de lo contrario, se los establece en 1. Ahora la nueva matriz solo posee valores binarios, estos valores serán tomados en orden de las manecillas del reloj obteniendo un número binario el cual será representado en forma decimal en la celda central o de umbral. Esto se observa de mejor manera en la Figura 1.28.

Se procede a realizar el mismo procedimiento para todos los píxeles de la imagen de entrada. Al final de este procedimiento, se tiene una nueva imagen que representa mejor las características de la imagen original.

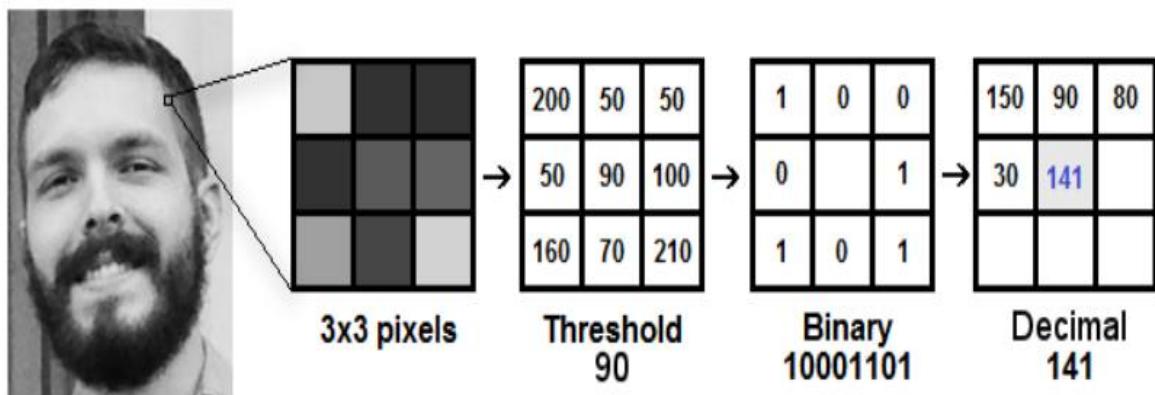


Figura 1.28. Operación LBP aplicada a una imagen [49].

En la Figura 1.29, se puede observar la imagen después de haberla procesado con el algoritmo LBP.

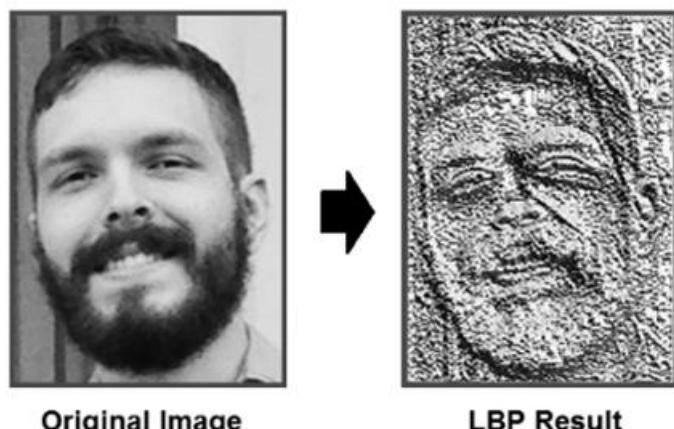


Figura 1.29. Imagen orinal (izquierda) e Imagen procesada con LBP (derecha) [49].

Con las imágenes de entrenamiento procesadas con el algoritmo LBP y con la imagen de entrada procesada, también, con el algoritmo LPB ya se procede con la comparación. Cada pixel de las imágenes procesadas representa un histograma; para realizar un reconocimiento facial se debe comparar el histograma de la imagen de entrada con los histogramas de las imágenes de entrenamiento.

Finalmente, un sistema de reconocimiento facial posee varias ventajas y desventajas, las cuales se mencionan a continuación:

Ventajas:

- No es intrusiva. El reconocimiento facial se hace a la distancia. A unos pocos metros de la cámara de video.
- Existen muchos productos comerciales que ofrecen esta tecnología a un bajo precio.
- No hay variaciones, significativas, en la estructura de la cara con el pasar de los años.
- No afecta el hecho de usar maquillaje excesivo.
- Se realiza reducción de dimensiones para extraer las características de la persona.

Desventajas:

- Debe poseer condiciones de iluminación adecuadas al sistema de reconocimiento facial.
- Unidades gubernamentales pueden acceder a los datos biométricos para dar seguimiento a las personas en cualquier parte del mundo.

1.3.8. HUELLA DACTILAR

El método de huella dactilar para sistemas biométricos, es uno de los más estudiados y uno de los más implementados a nivel mundial; ya que posee una fiabilidad muy alta, es fácil de usar y tiene una amplia aceptación por parte de los usuarios, debido a que existen numerosos estudios que avalan la unicidad de la huella dactilar de los individuos, sin olvidar que ésta no cambia con la edad de las personas.

Las huellas dactilares están constituidas por rugosidades de la piel que forman salientes y depresiones [16]. Para la extracción de las huellas dactilares se debe tener en consideración las características de las huellas dactilares, las más importantes son:

- Crestas (Rides): son la parte más sobresaliente, o elevada, de la huella dactilar. Cuando se imprime la huella dactilar las crestas vienen a ser las zonas o rayas negras.
- Valles (Valleys): son la parte profunda de la huella dactilar, esta se encuentra entre los valles de la huella dactilar. Cuando se imprime la huella dactilar los valles vienen a ser las zonas blancas.
- Curvas (Loops): es el sector donde una cresta toma una curva, generalmente en forma de "U". Se lo conoce, también, como lazo.
- Bifurcaciones (Deltas): es donde la cresta en cualquier parte de su recorrido se divide en dos crestas que continúan paralelamente. También es la parte de la huella dactilar donde hay una triangulación.
- Espirales (Whorls): como el nombre lo indica, es una cresta tipo curva, forma una espiral.
- Núcleo (Core): es el centro de la huella dactilar, es donde se genera el inicio del lazo en la huella o el inicio de una espiral.
- Terminación (Ending): es donde la cresta termina y no continua. Esta se ubica entre dos crestas.
- Minucias: es el punto de interés de la huella dactilar, estos puntos pueden ser utilizados para el reconocimiento dactilar de una persona [18].

En la Figura 1.30, se puede observar algunas de las características de las huellas dactilares como son: los valles, las crestas, los lazos, las bifurcaciones o deltas, el espiral y el núcleo de la huella dactilar.

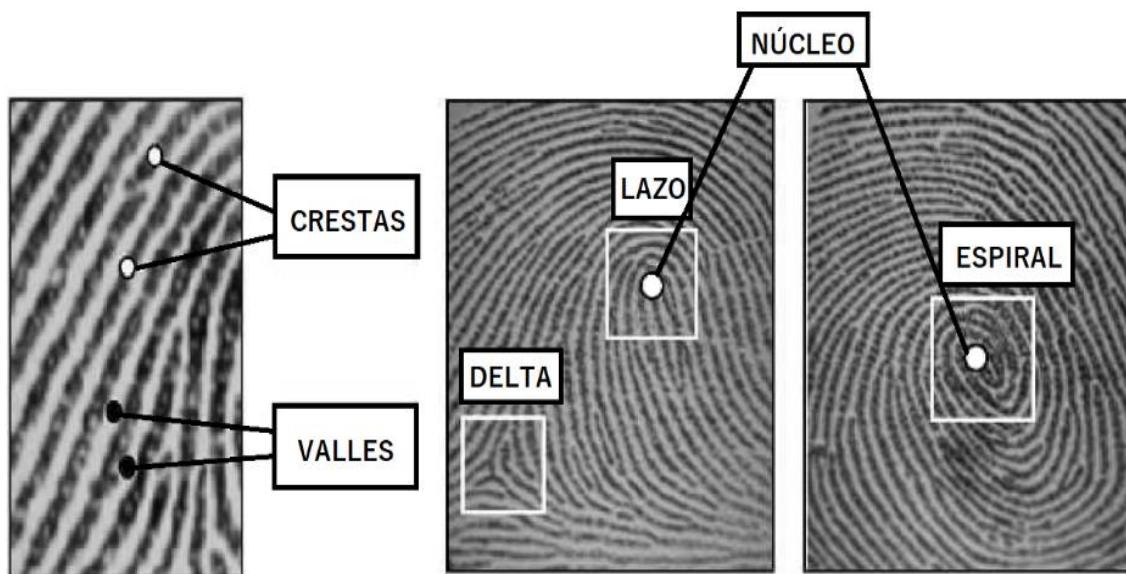


Figura 1.30. Algunas de las características de una huella dactilar [19].

La terminación de una cresta, así como una serie de minucias se pueden observar en la Figura 1.31.

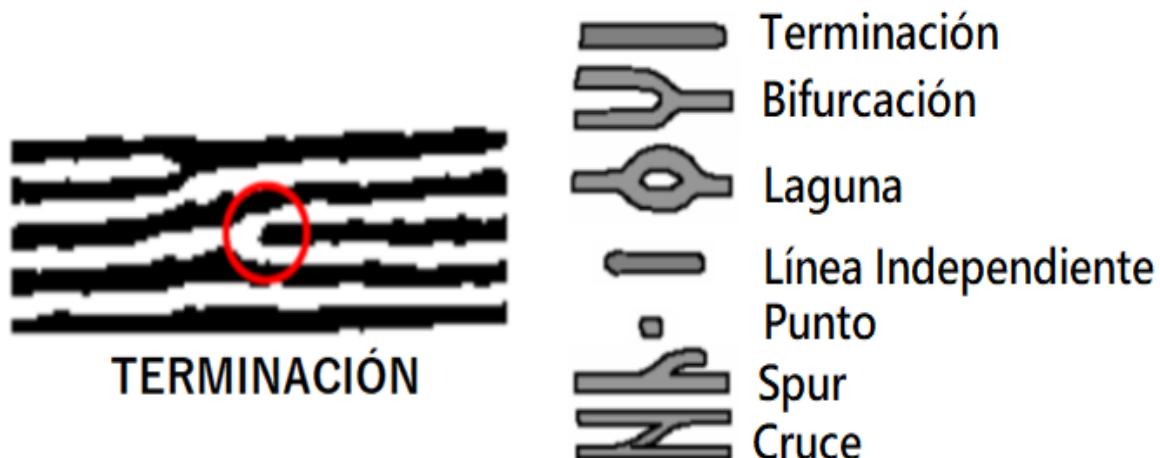


Figura 1.31. Diferentes tipos de minucias [18].

Todas las características antes nombradas conforman el patrón de la huella dactilar, este patrón es único incluso entre gemelos idénticos; se estima que la probabilidad de que dos personas tengan la misma huella dactilar es de 1 en 64.000 millones.

Existe una clasificación para las huellas dactilares, uno de los métodos más comunes para clasificar es el método Henry; este método clasifica las huellas dactilares en cinco grupos, como se muestra en la Figura 1.32.



Figura 1.32. Diferentes tipos de patrón de huellas dactilares [22].

Para realizar un reconocimiento de una huella dactilar, primero, se debe digitalizar la huella; esto permite obtener las características más relevantes, las minucias, de una huella. Para digitalizar una huella se hace uso de algoritmos que permiten obtener un índice numérico correspondiente a la huella dactilar; cuando una persona desea ser identificada, solo debe poner su dedo sobre el lector dactilar, el mismo que escanea la huella para luego ser comparada con una almacenada en una base de datos.

Las técnicas más utilizadas para la extracción de las huellas dactilares son:

- Segmentación: la segmentación es una tarea de procesamiento de imágenes la cual consiste en separar el área de la huella dactilar del fondo.
- Mejoramiento y Binarización: el objetivo de esta técnica es mejorar la claridad de la estructura de las crestas.
- Técnicas Basadas en Correlación: mediante la utilización de esta técnica se analiza el patrón global seguido por la huella dactilar, es decir, el esquema general del conjunto de la huella en lugar de las minucias.
- Basada en minucias: esta técnica basa su mecanismo de autenticación en determinadas formas fácilmente identificables existentes en la huella dactilar. Así, se registra el tipo de minucia y su posición dentro de la huella, estableciendo una serie de mediciones. De esta forma, el modelo o plantilla correspondiente a cada usuario es un esquema en el que se indican las minucias que se han de detectar, su posición y las distancias que separan unas de otras [22].

Para poder reconocer a una persona mediante su huella dactilar, se debe proceder a extraer las minucias de una huella de entrada; una vez extraídas las minucias éstas son realizadas, luego son medidas en ubicación y distancia entre ellas. Finalmente, es

trasformada en una plantilla la cual es almacenada o comparada con una ya existente en una base de datos. Se puede observar de mejor manera en la Figura 1.33.

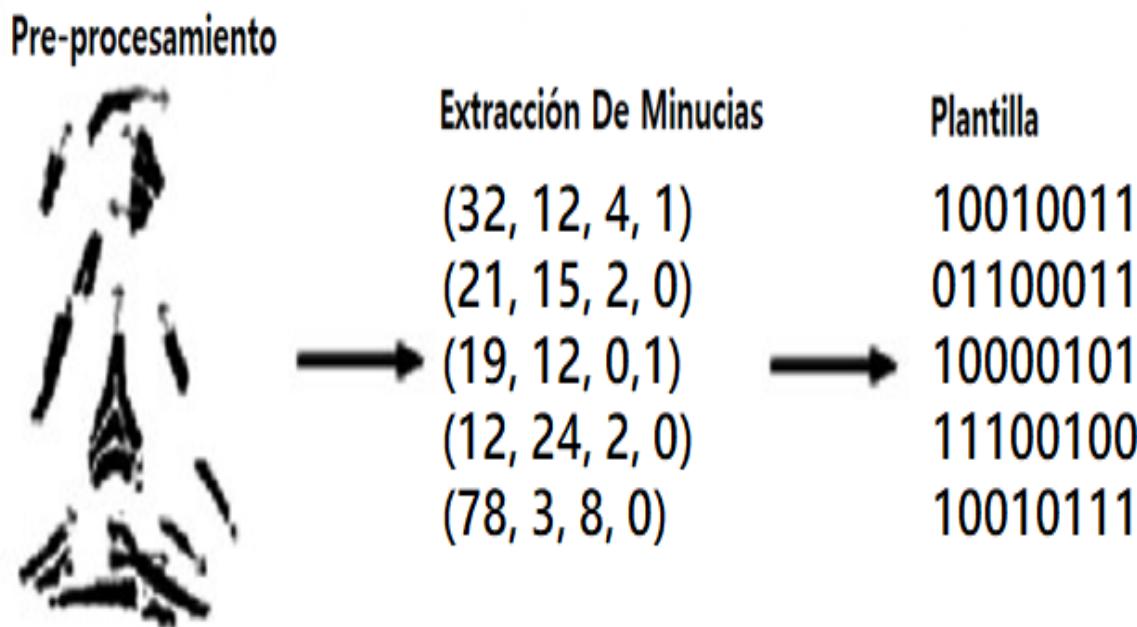


Figura 1.33. Procesamiento de una huella dactilar [16].

Dentro de la utilidad de usar un sistema biométrico basado en reconocimiento dactilar, se puede nombrar algunas de las ventajas y desventajas que tiene este sistema.

Ventajas:

- Es uno de los sistemas más seguros que existe, con índice de acierto del 99%.
- La información en las huellas dactilares es significativa. Las huellas de cada uno de dedos de la mano de una persona son diferentes; por lo cual se puede usar cualquier dedo para identificar a una persona.
- Las huellas dactilares no pueden ser transferidas a otro individuo.

Desventajas:

- No todas las personas del mundo tienen huellas dactilares. Existe un grupo reducido de personas que no posee huellas dactilares, a esto se le conoce como “Adermatoglifia”. Por lo cual no todos pueden acceder a este método de identificación.
- El sensor es vulnerable a ataque físico, lo que impediría que las personas que desean identificarse no puedan hacerlo.
- Enfermedades a la piel, como la dermatitis, o daño a los dedos pueden entorpecer el sistema biométrico dactilar.

1.3.9. METODOLOGÍAS DE DESARROLLO DE SOFTWARE

Las metodologías de desarrollo software son marcos de trabajo usadas para estructurar, planificar y controlar el proceso de desarrollo en sistemas de información; existen dos tipos de metodologías, las llamadas tradicionales o clásicas y las metodologías agiles.

Las primeras son un tanto rígidas, exigen una documentación exhaustiva y se centran en cumplir con el plan del proyecto definido totalmente en la fase inicial del desarrollo del mismo; mientras que la segunda enfatiza el esfuerzo en la capacidad de respuesta a los cambios, las habilidades del equipo y mantener una buena relación con el usuario. A continuación, se muestra la Tabla 1.5 donde se pueden observar algunas de las metodologías existentes [21].

Tabla 1.5. Metodologías para el desarrollo de software.

METODOLOGÍAS DE DESARROLLO SOFTWARE	
Metodologías Clásicas	Metodologías Ágiles
RUF	Programación Extrema o XP
MSF	Scrum
Iconix	Crytal
Win-Win Spiral Model	Mobile – D
Desarrollo Basado En Componentes	Desarrollo de Software Adaptativo
	Kanban

1.3.10. METODOLOGÍA DE DESARROLLO SOFTWARE KANBAN

La expresión Kanban, que en japonés significa “tarjeta” o “registro visible”, se refiere a las tarjetas que se utilizan para controlar el flujo de producción en la fábrica [9]. Kanban es un método que pone énfasis en la entrega del producto justo a tiempo sin realizar sobrecarga al trabajo de los miembros del equipo; esto quiere decir que Kanban posee un sistema que le indica qué producir, cuando producir y cuánto producir.

Kanban tiene ciertas ventajas sobre otras metodologías de desarrollo de software, estas ventajas son la división del trabajo en tareas como se muestra a continuación:

- Lista de Tareas: acciones por realizarse en el proyecto.
- Tareas en proceso: acciones que se están desarrollando en el proyecto.

- Tareas Realizadas: acciones finalizadas del proyecto.

Aunque Kanban es útil para el desarrollo de software, también posee ciertas desventajas, estas son:

- No es útil para proyectos grandes y tiempos muy definidos.
- No controla los tiempos dedicados.
- No existe feedback de las tareas realizadas.

Para trabajar con la metodología Kanban solo es necesario poseer una pizarra, cartel o cualquier objeto visual donde los miembros del equipo de trabajo puedan apreciar las tareas por realizar, las tareas que se están realizando y las tareas que finalizaron, así se puede ver en la Figura 1.34.

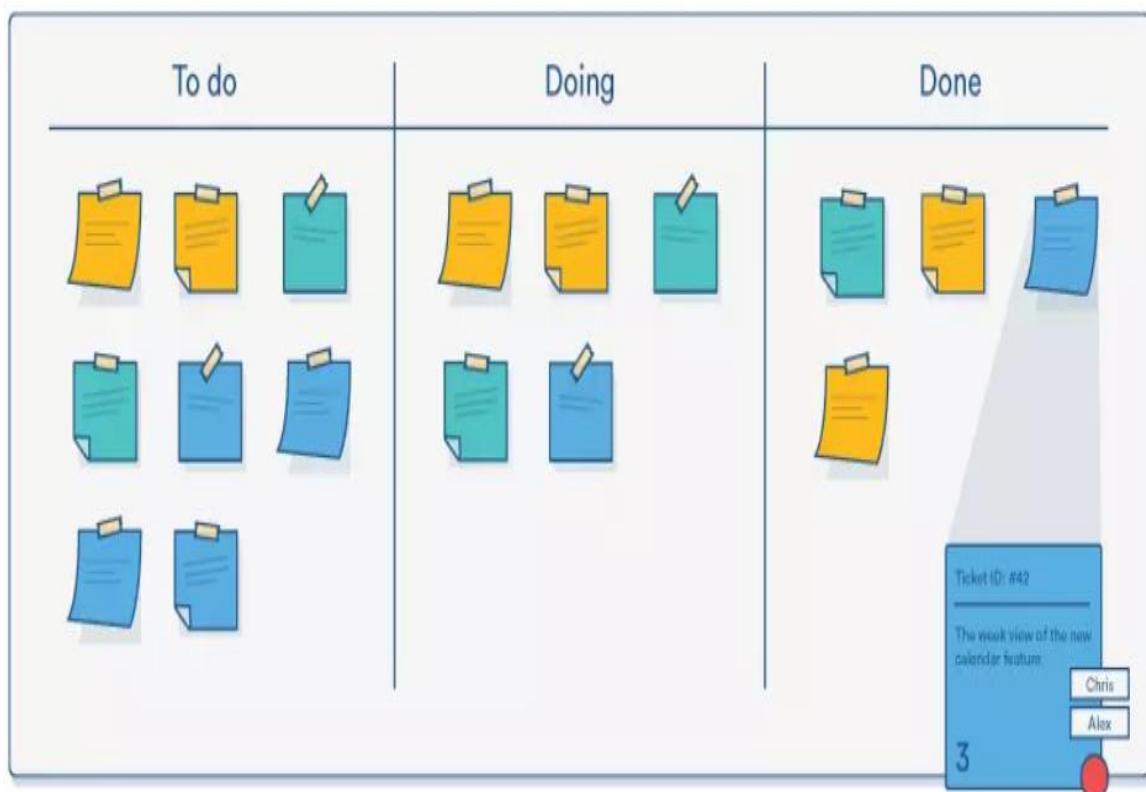


Figura 1.34. Tablero Kanban [50].

1.3.11. SOFTWARE Y HARDWARE

Para el desarrollo de este proyecto será necesario el uso de varios elementos hardware y software; éstos se muestran en la Tabla 1.6 para los elementos hardware y en la Tabla 1.7 se muestran los elementos software.

Tabla 1.6. Elementos hardware a utilizar.

HARDWARE	
ELEMENTO	DESCRIPCIÓN
Computador	Se puede hacer uso de un computador de escritorio o portátil que cumpla con las características software mencionadas en la Tabla 1.7.
Cámara IP	Se puede hacer uso de cualquier cámara IP, siempre, que sea compatible con Onvif.
Lector dactilar	Los lectores dactilares que se pueden usar para el desarrollo de este proyecto son ZK9500 o ZK4500.
Panel numérico	Cualquier Panel numérico, en buenas condiciones, es útil.
Cable UTP con interfaz física RJ45	El Cable UTP debe ser de la categoría 5e con la norma EIA/TIA-568-B.
Router	Es recomendable usar un router para la asignación dinámica de la IP a las cámaras. El router puede ser de cualquier marca, pero es recomendable que sea de los más actuales para un rápido streaming de video.
USB Hub	Se puede hacer uso de cualquier usb hub, se recomienda hacer uso de uno de mínimo de 3 puertos y con velocidad de 2.0.
Cables USB	Si es necesario hacer extensiones de los dispositivo se recomienda hacer uso de un cable USB Tipo A (Hembra y Macho) con velocidad mínima de 2.0.
Arduino	Se puede hacer uso de cualquier modelo de Arduino. Se recomienda hacer uso del Arduino Uno ya que no se requiere de gran proceso por parte de este.
Cerradura Magnética.	Las cerraduras magnéticas que se pueden usar pueden ser de cualquier modelo. Es preferible usar cerraduras que soporten fuerzas mayores a 300 libras.
Otros.	Cables conectores para Arduino, extensiones eléctricas, transformadores de 12v DC, leds, relés, resistencias y canaletas.

Tabla 1.7. Elementos software a utilizar

SOFTWARE	
ELEMENTO	DESCRIPCIÓN
Sistema Operativo Windows	Se puede hacer uso del Sistema Operativo Windows en sus versiones 7, 8 y 10; aunque para el correcto funcionamiento se recomienda hacer uso del Sistema Operativo Windows 10.
Visual Studio 2017	Visual Studio, OpenCV, EmguCV y Ozeki ya fueron explicados en sus respectivos apartados. Cabe mencionar que se puede usar cualquier versión de estos software; pero es recomendable usar las versiones más recientes.
OpenCV (Librerías)	Visual Studio, OpenCV, EmguCV y Ozeki ya fueron explicados en sus respectivos apartados. Cabe mencionar que se puede usar cualquier versión de estos software; pero es recomendable usar las versiones más recientes.
EmguCV (Librerías)	
Ozeki (Librerías)	

2. METODOLOGÍA

Se dividirá este capítulo en dos apartados, el primero de ellos será el diseño del sistema prototipo para el acceso a los Laboratorios de Redes de la FIEE de la EPN basado en reconocimiento facial; donde se definirá los requisitos del sistema prototipo, se realizarán diagramas de actividades, casos de uso, entidad relación y diagramas de clases de los módulos del aplicativo. El segundo apartado, será de la implementación del sistema prototipo, tanto en la parte física como la lógica respetando lo establecido en el primer apartado.

2.1. DISEÑO DEL SISTEMA PROTOTIPO

La finalidad del proyecto a realizar es el de la implementación de un sistema de seguridad (software y hardware) que permita reconocer a una persona mediante el uso de biométricos o un PIN de acceso; para realizar el diseño del sistema prototipo se seguirán los siguientes pasos:

- Se seleccionará la metodología de desarrollo de software a utilizarse.
- Se verificará la situación actual y las necesidades presentes en el Laboratorio de Redes de la FIEE de la EPN.
- Se determinarán los requerimientos funcionales y no funcionales.
- Se elaborarán los diagramas de casos de usos y secuencia para el uso de la aplicación del sistema prototipo.
- Se realizarán los diagramas de clase de los módulos: “Interfaz de Usuario”, “Lógica de Negocios” y “Acceso a Datos”.
- Se realizarán los diagramas de actividades de cada módulo: “Interfaz de Usuario”, “Lógica de Negocios” y “Acceso a Datos”.
- Se realizarán los diagramas entidad relación del módulo “Base de Datos”.
- Se determinarán los requerimientos físicos del sistema de seguridad.

2.1.1. SELECCIÓN METODOLOGÍA DE DESARROLLO SOFTWARE KANBAN

La metodología de desarrollo de software Kanban fue seleccionada debido a su facilidad de uso, como se explica en los capítulos 1.3.9 y 1.3.10. En la Tabla 2.1 se detalla una columna llamada “Tareas Por Realizar” esta es una lista de actividades correspondientes

a las fases de diseño, implementación y pruebas. No existen “Tareas en Proceso” y “Tareas Finalizadas”.

Tabla 2.1. Tablero Kanban fase inicial de diseño.

TABLERO KANBAN		
TAREAS POR REALIZAR	TAREAS EN PROCESO	TAREAS FINALIZADAS
Selección metodología de Desarrollo Software.		
Análisis situación actual laboratorios de la FIEE.		
Encuesta de requerimientos.		
Diagramas de casos de uso.		
Diagramas de secuencia.		
Diagramas de clase.		
Diagramas de actividades.		
Diagramas entidad relación.		
Requerimientos físicos del sistema de seguridad.		
Instalación Visual Studio 2017.		
Instalación Librerías Ozeki, EmguCV, OpenCV, ZKTeco e iText.		
Creación de los formularios de la interfaz gráfica.		
Implementación de cada módulo.		
Configuración de los terminales de acceso.		
Configuraciones de equipos intermedios.		
Ubicación física de los terminales de acceso.		
Extensión del cableado necesario para la comunicación entre los equipos.		
Comprobaciones de funcionamiento de los módulos.		
Correcciones de los módulos.		
Verificar la existencia de errores producidos al interconectar los equipos.		
Depurar los errores producidos al interconectar los equipos.		
Probar el funcionamiento de las cámaras, lectores de huellas y pines de acceso numérico.		
Se realizarán las encuestas de satisfacción.		

2.1.2. SITUACIÓN ACTUAL Y LAS NECESIDADES PRESENTES EN EL LABORATORIO DE REDES DE LA FIEE DE LA EPN

Actualmente la EPN no cuenta con un sistema de seguridad electrónico para el ingreso al campus o a las diferentes áreas de la EPN como pueden ser los laboratorios, aulas, oficinas, etc.; por lo que la EPN está expuesta al ingreso de todo tipo de personas, que amenazan la seguridad de las personas y de los diferentes equipos que se encuentra en el interior de la EPN.

La EPN es vulnerable a robos hacia los estudiantes, trabajadores, docentes y personal administrativo; y no solo eso, los laboratorios donde se encuentran equipos, materiales e instrumentos de trabajo, que son costosos y de difícil reposición, además se encuentran expuestos.

Si bien existen algunos esfuerzos por asegurar a la comunidad politécnica, los laboratorios, aulas, oficinas y bienes de la EPN con guardias, puertas metálicas y candados; esto no resultaría mayor problema para ladrones experimentados que podrían burlar estas seguridades con relativa facilidad y al no contar con un sistema de seguridad que salvaguarde la integridad de estudiantes, trabajadores, docentes, personal administrativo, equipos, materiales e instrumentos de trabajo y que permita identificar a las personas que circulan por las diferentes áreas de la EPN, los perjuicios hacia la institución y los que la integran seguirán sucediendo e incrementándose con el tiempo.

La integración de un sistema de seguridad electrónico, especialmente, en los laboratorios de la EPN donde se encuentran equipos costosos, ayudaría a mejorar la seguridad de la institución; ya que este sistema contará con control de acceso biométrico y cámara de seguridad que permitirán identificar al personal que ingrese a los laboratorios de la EPN.

2.1.3. REQUERIMIENTOS FUNCIONALES Y NO FUNCIONALES

Los requerimientos funcionales tratan de establecer lo que el sistema debe hacer, sus propiedades y las restricciones en el funcionamiento del sistema. La toma de los requerimientos funcionales y los requerimientos no funcionales se la realizó mediante una encuesta realizada al personal de la FIEE de la EPN; cuyas actividades dentro de los laboratorios es la más relevante, lo cual permitiría tomar decisiones más acertadas al

exponer las principales necesidades de los laboratorios respecto a la seguridad de los mismos.

En las Tablas 2.2 y 2.3 se exponen los requerimientos funcionales y los requerimientos no funcionales obtenidos. La información sobre las encuestas se encuentra en el “Anexo C”.

Tabla 2.2. Requerimientos funcionales.

REQUERIMIENTOS FUNCIONALES.	
RF	DESCRIPCIÓN
RF001	Detección y reconocimiento dactilar.
RF002	Detección y reconocimiento facial.
RF003	Acceso mediante PIN.
RF004	Los administradores podrán ingresar al sistema por medio de contraseñas únicas.
RF005	Selección por el administrador del sistema de acceso a utilizar.
RF006	Buscar cámaras, lectores dactilares y teclados numéricos conectados.
RF007	Selección de la sala de la cual se desea conectar o desconectar los dispositivos.
RF008	Añadir, modificar y eliminar usuarios de la base de datos.
RF009	Mostrar en pantalla la cámara para la prueba de software.
RF010	Tomar y modificar imágenes o video de la cámara.
RF011	Mostrar en pantalla el lector dactilar para la prueba de software.
RF012	Mostrar en pantalla el teclado numérico para la prueba de software.
RF013	Mostrar imagen ya entrenada (blanco y negro).
RF014	Mostrar información de los dispositivos conectados.
RF015	Mostrar en pantalla informe de los eventos ocurridos con el uso del sistema.
RF016	El sistema no debe poseer más de dos administradores.
RF017	Las contraseñas deben ser alfanuméricas y con una longitud determinada.
RF018	No puede haber usuarios repetidos.
RF019	Generar un documento con los eventos totales o parciales.

Tabla 2.3. Requerimientos no funcionales.

REQUERIMIENTOS NO FUNCIONALES	
RNF	DESCRIPCIÓN
RNF001	Diseño de una interfaz sencilla, amigable e intuitiva para su fácil utilización.
RNF002	La base de datos al implementarla de forma local siempre estará disponible.
RNF003	El sistema prototipo debe soportar a varios usuarios simultáneos, al menos dos (uno por cada sala).
RNF004	El sistema prototipo será desarrollado en una plataforma que permita modificar el software a largo plazo.
RNF005	Completa documentación del código comentado.
RNF006	Integridad de las imágenes, huellas y PIN en la base de datos.
RNF007	Para el registro de usuarios se requiere la presencia física del usuario.
RNF008	Los dispositivos de acceso estarán al alcance de los usuarios.

En la Tabla 2.4 se muestra la primera actualización del tablero Kanban, la columna llamada “Tareas En Proceso” corresponden a tareas de la fase de diseño y que se encuentran en etapa de desarrollo; finalmente, la columna llamada “Tareas Finalizadas” son las actividades finalizadas del proyecto.

Tabla 2.4. Primera actualización tablero Kanban.

TABLERO KANBAN		
TAREAS POR REALIZAR	TAREAS EN PROCESO	TAREAS FINALIZADAS
Instalación Visual Studio 2017.	Diagramas de casos de uso.	Selección metodología de Desarrollo Software.
Instalación Librerías Ozeki, EmguCV, OpenCV, ZKTeco e iText.	Diagramas de secuencia.	Ánálisis situación actual laboratorios de la FIEE.
Creación de los formularios de la interfaz gráfica.	Diagramas de clase.	Encuesta de requerimientos.
Implementación de cada módulo.	Diagramas de actividades.	
Configuración de los terminales de acceso.	Diagramas entidad relación.	
Configuraciones de equipos intermedios.	Requerimientos físicos del sistema de seguridad.	
Ubicación física de los terminales de acceso.		
Extensión del cableado necesario para la comunicación entre los equipos.		
Comprobaciones de funcionamiento de los módulos.		
Correcciones de los módulos.		
Verificar la existencia de errores producidos al interconectar los equipos.		
Depurar los errores producidos al interconectar los equipos.		
Probar el funcionamiento de las cámaras, lectores de huellas y pines de acceso numérico.		
Se realizarán las encuestas de satisfacción.		

2.1.4. DIAGRAMAS DE CASOS DE USOS Y SECUENCIA PARA EL USO DE LA APLICACIÓN DEL SISTEMA PROTOTIPO

Los siguientes diagramas de casos de uso muestran la interacción de los diferentes usuarios con el sistema prototipo de seguridad y sus diferentes funcionalidades. En la Figura 2.1 se puede observar una visión general del uso de la aplicación, donde el usuario administrador es el que inicia la aplicación y una vez iniciada la aplicación puede realizar tareas como: agregar, modificar y eliminar usuarios, entre otras acciones. El usuario final es el encargado de proporcionar los datos para el registro o modificación y es el encargado de utilizar los dispositivos de acceso.

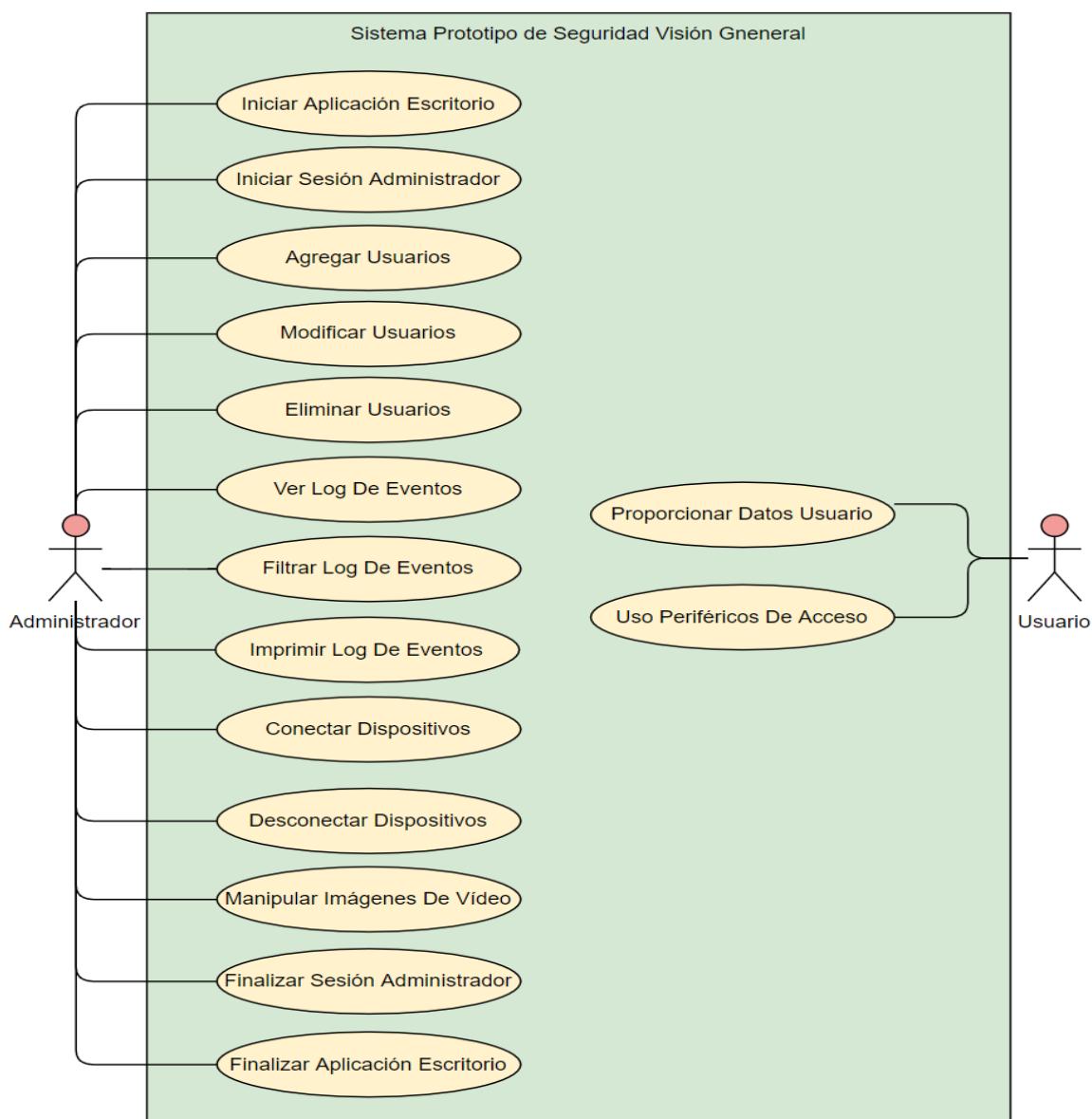


Figura 2.1. Diagrama de caso de uso general.

Se podrá hacer uso de la aplicación si el sistema operativo cumple con los requisitos mínimos de operación.

Una vez iniciada la aplicación el usuario administrador tendrá que iniciar sesión; para ello dentro del formulario “Inicio_Sesion” deberá proporcionar sus credenciales, las cuales serán comprobadas en la base de datos; si el inicio de sesión es exitoso se podrá proceder con la administración de la aplicación.

En la Figura 2.2 se puede apreciar el caso de uso y en la Figura 2.3 el diagrama de secuencia para el administrador en el caso de iniciar sesión.

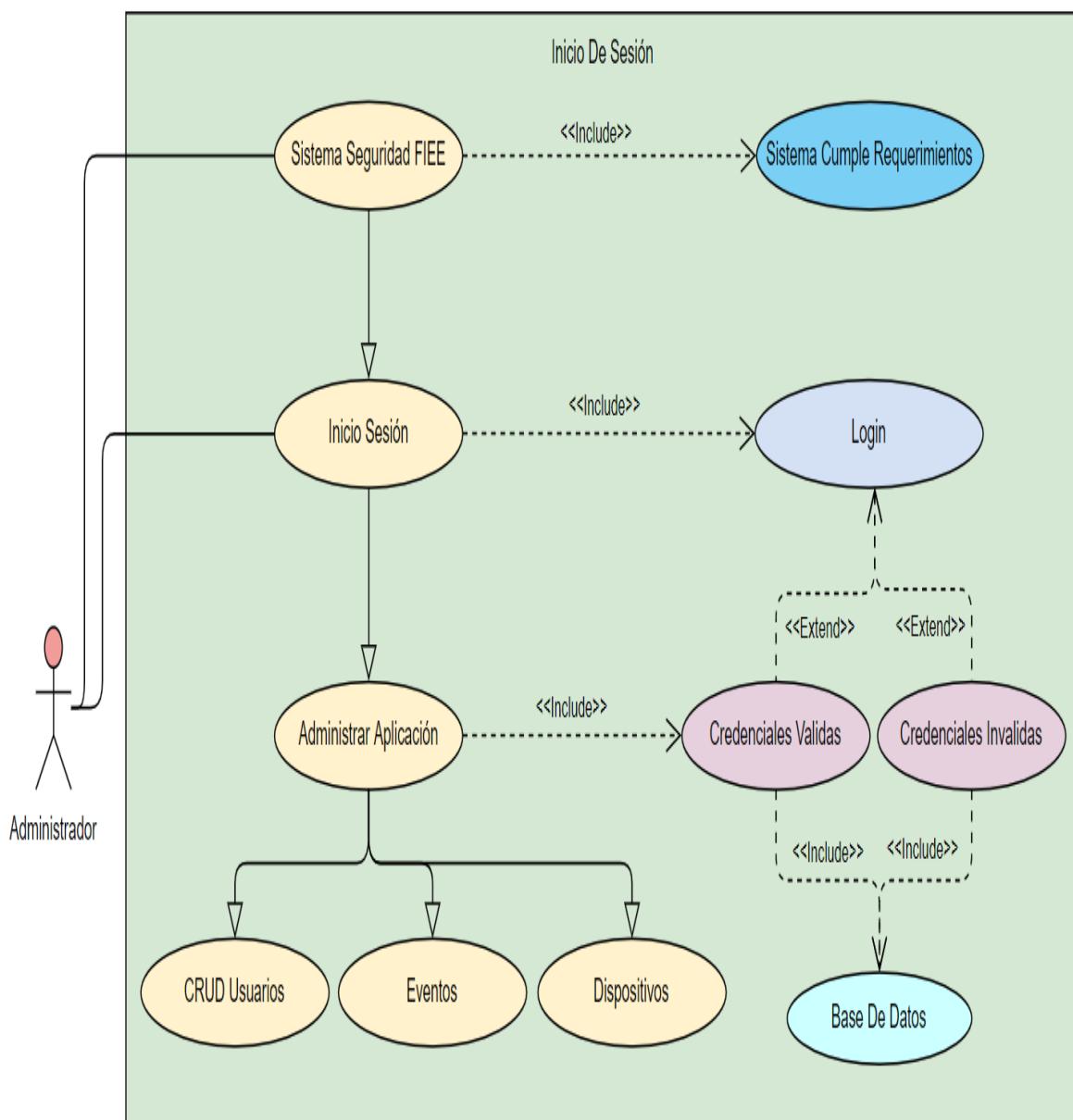


Figura 2.2. Diagrama de caso de uso del inicio de sesión.

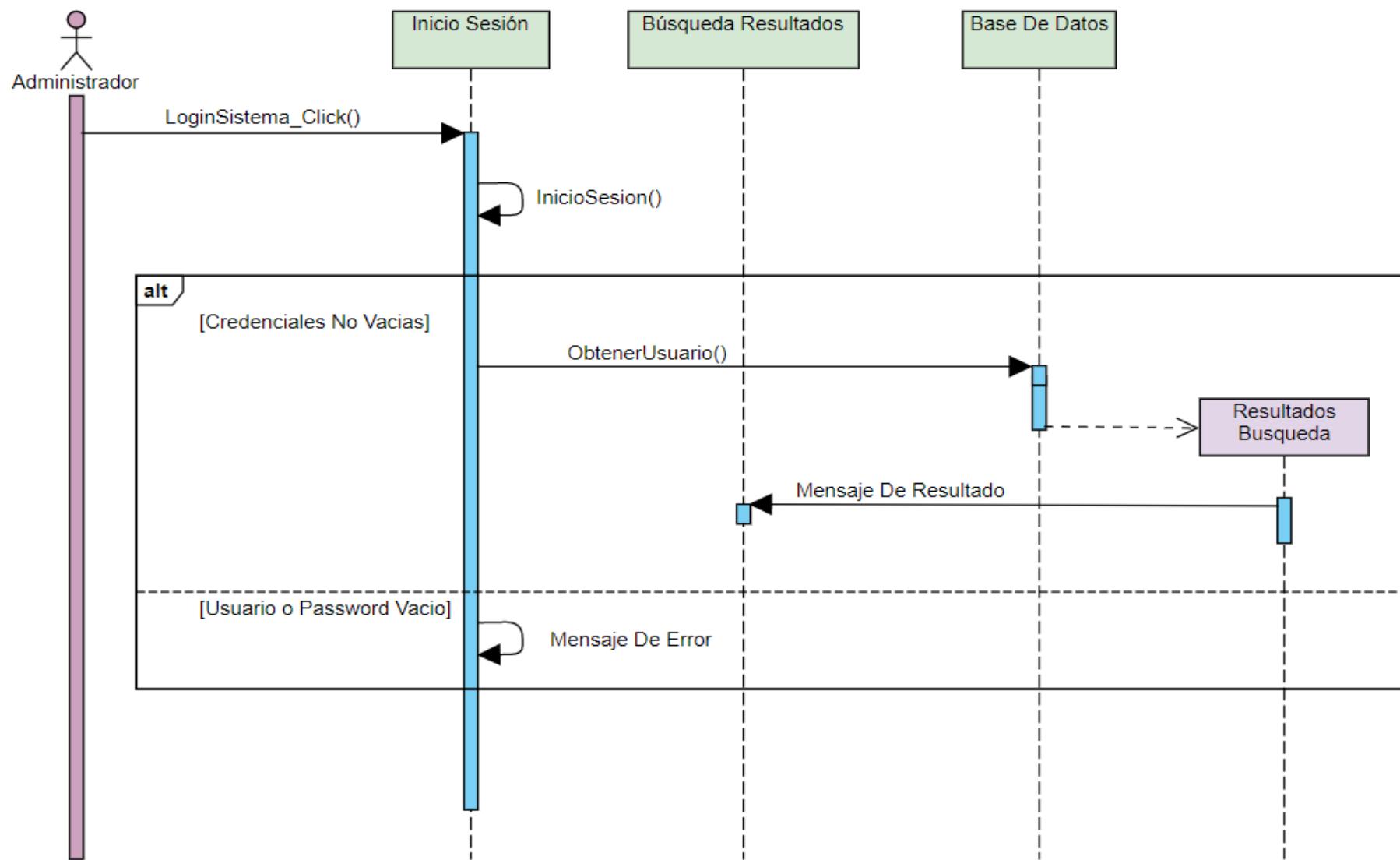


Figura 2.3. Diagrama de secuencia del inicio de sesión.

Una vez iniciada la sesión, el usuario administrador está en la facultad de agregar, modificar y eliminar usuarios del sistema; para ello, el administrador deberá ingresar al formulario “Administracion_Usuarios”.

Para el registro y actualización de los usuarios, el administrador necesitará de la colaboración de los usuarios finales; los cuales proporcionarán la información necesaria al administrador como: cédula, nombre, apellido, huella dactilar, pin de acceso, imagen del rostro, etc. Previo al registro o actualización de los usuarios, la información proporcionada por los usuarios finales será validada en la base de datos. En el caso de eliminación de usuarios, será el administrador el que decida qué usuario eliminar.

En la Figura 2.4 se puede observar el caso de uso para la administración (CRUD) de los usuarios por parte del administrador, y la Figura 2.5 se puede observar el registro y la modificación de un usuario; finalmente en la Figura 2.6 se encuentra la eliminación.

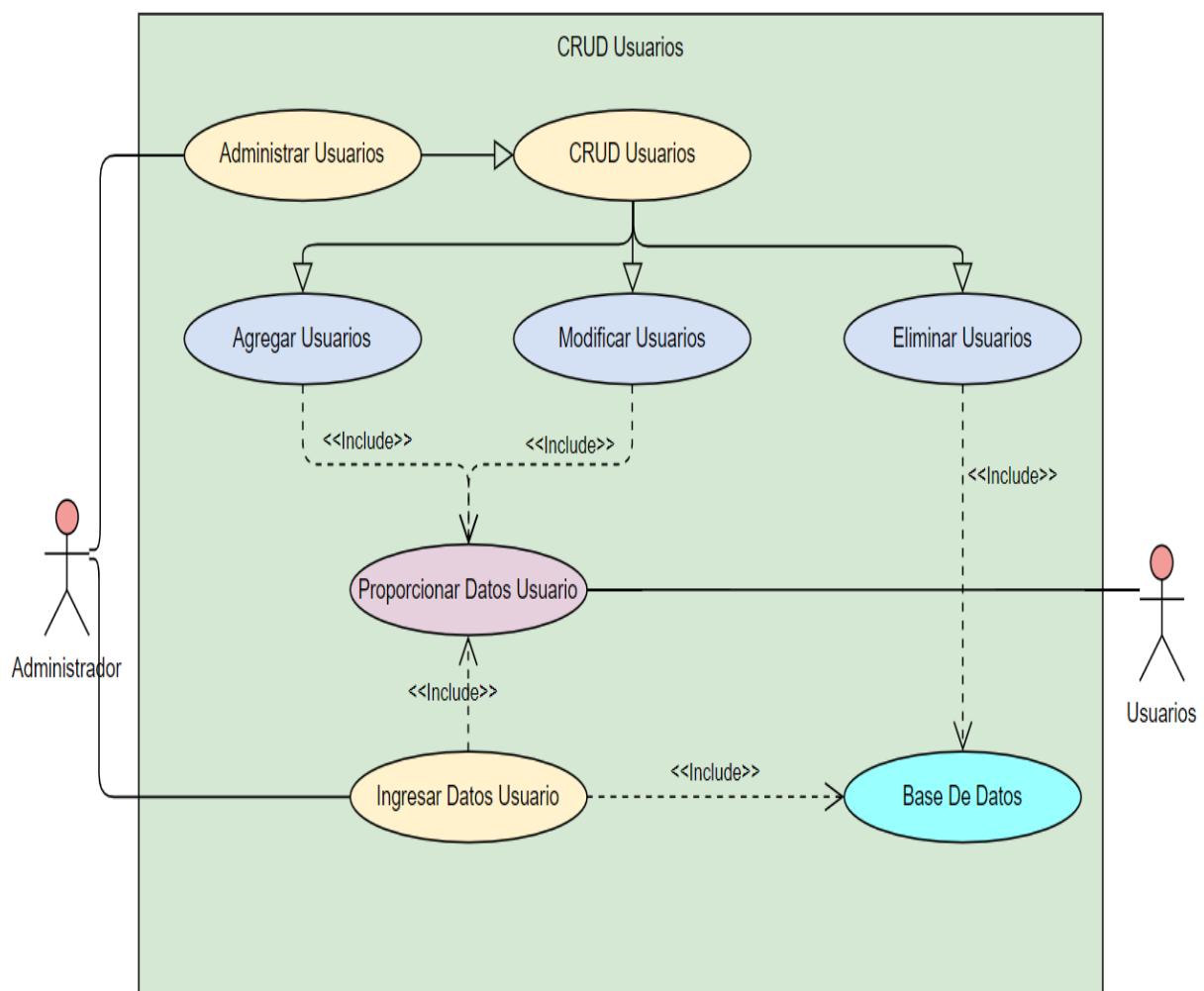


Figura 2.4. Diagrama de caso de uso del CRUD de usuarios.

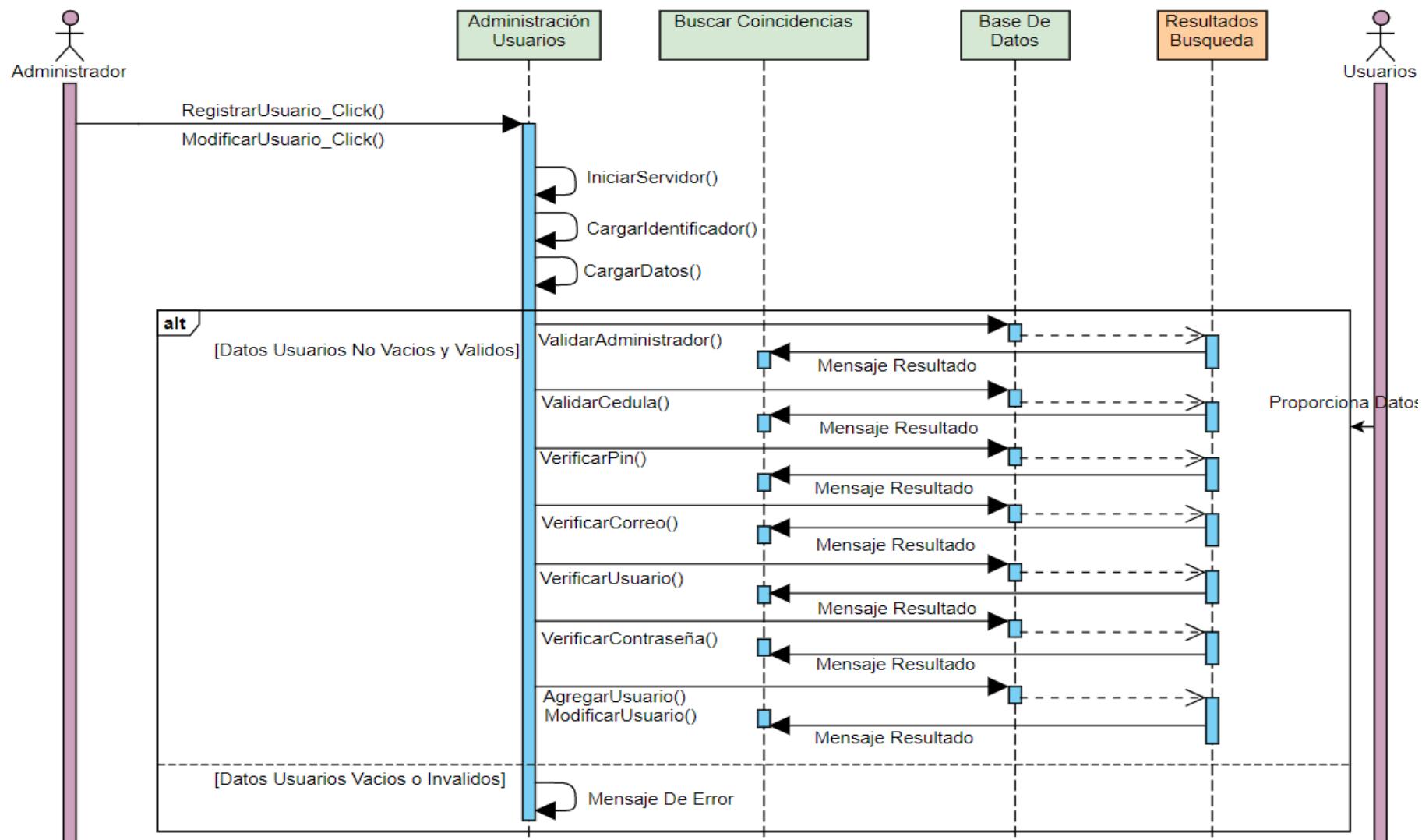


Figura 2.5. Diagrama de secuencia del registro y modificación de usuarios.

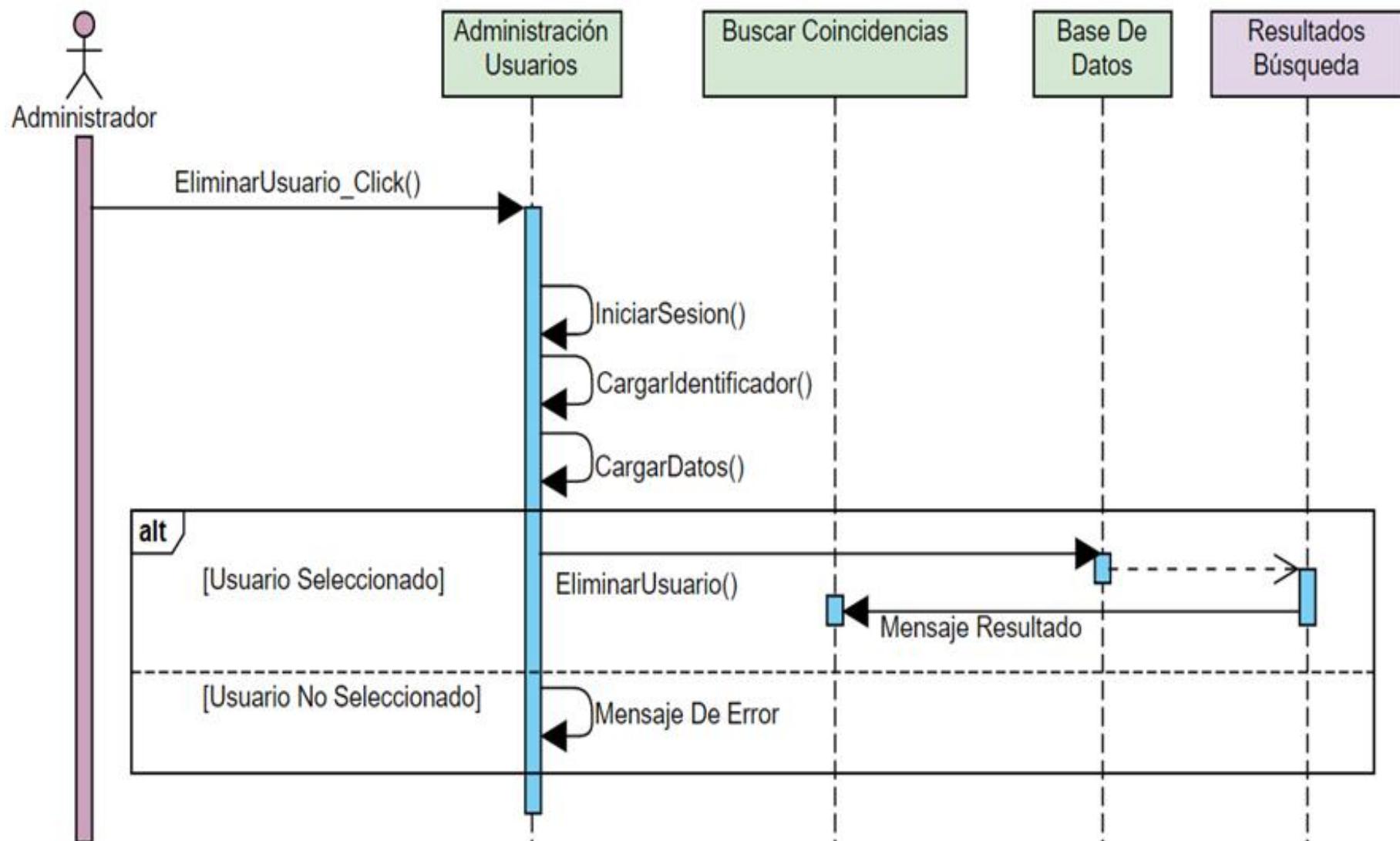


Figura 2.6. Diagrama de secuencia de la eliminación de usuarios.

La Figura 2.7 muestra el caso de uso para la visualización e impresión del “Log de Eventos” que se registran al hacer uso de la aplicación y en la Figura 2.8 se observa el diagrama de secuencia para la visualización e impresión del “Log de Eventos”.

Para este caso el administrador debe ingresar al formulario “Logs_Eventos”, donde todos los eventos son cargados por defecto; estos eventos pueden ser filtrados, mediante una cadena de búsqueda, si el administrador lo necesita. Los eventos podrán ser impresos con un nombre y formato especificados por el administrador en una ruta previamente seleccionada.

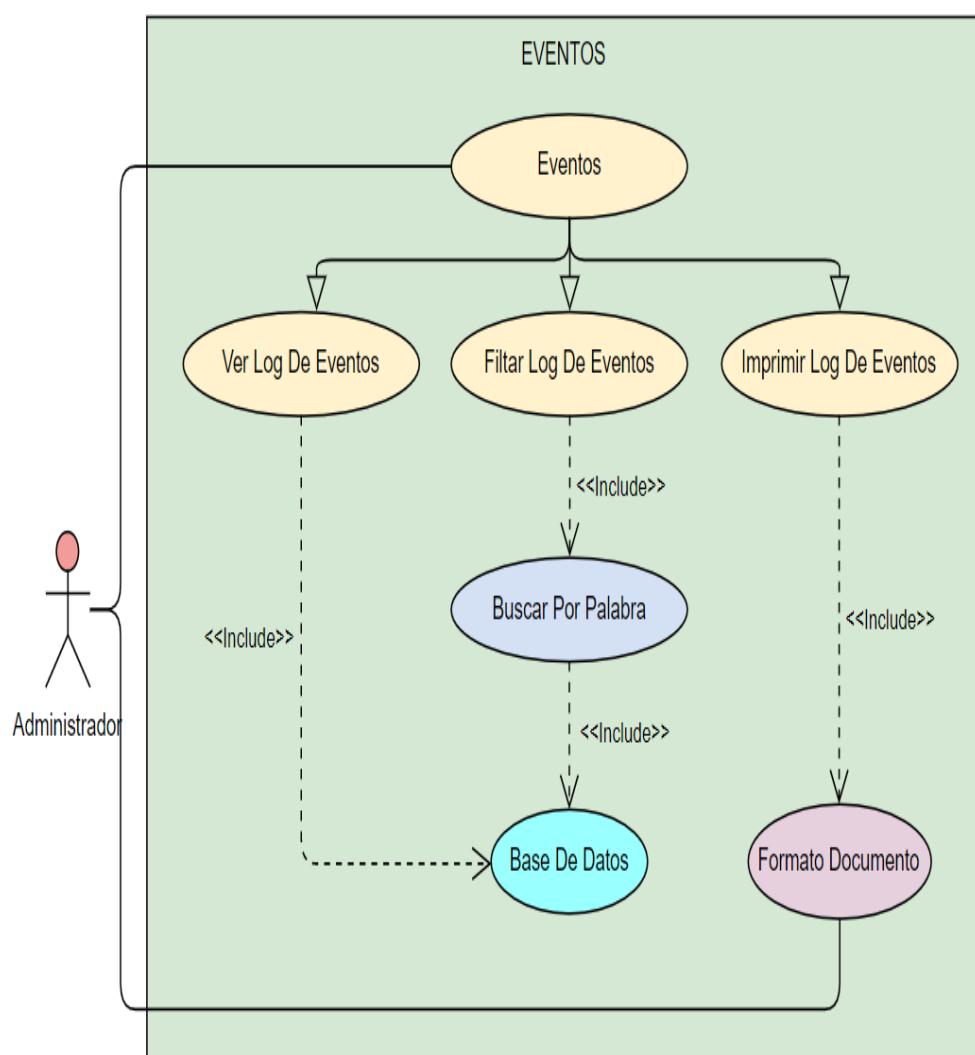


Figura 2.7. Diagrama de caso de uso administración del log de eventos.

El caso de uso para habilitar o deshabilitar los dispositivos periféricos se observa en la Figura 2.9. En la Figura 2.10 se observa el diagrama de secuencia para habilitar y deshabilitar dispositivos.

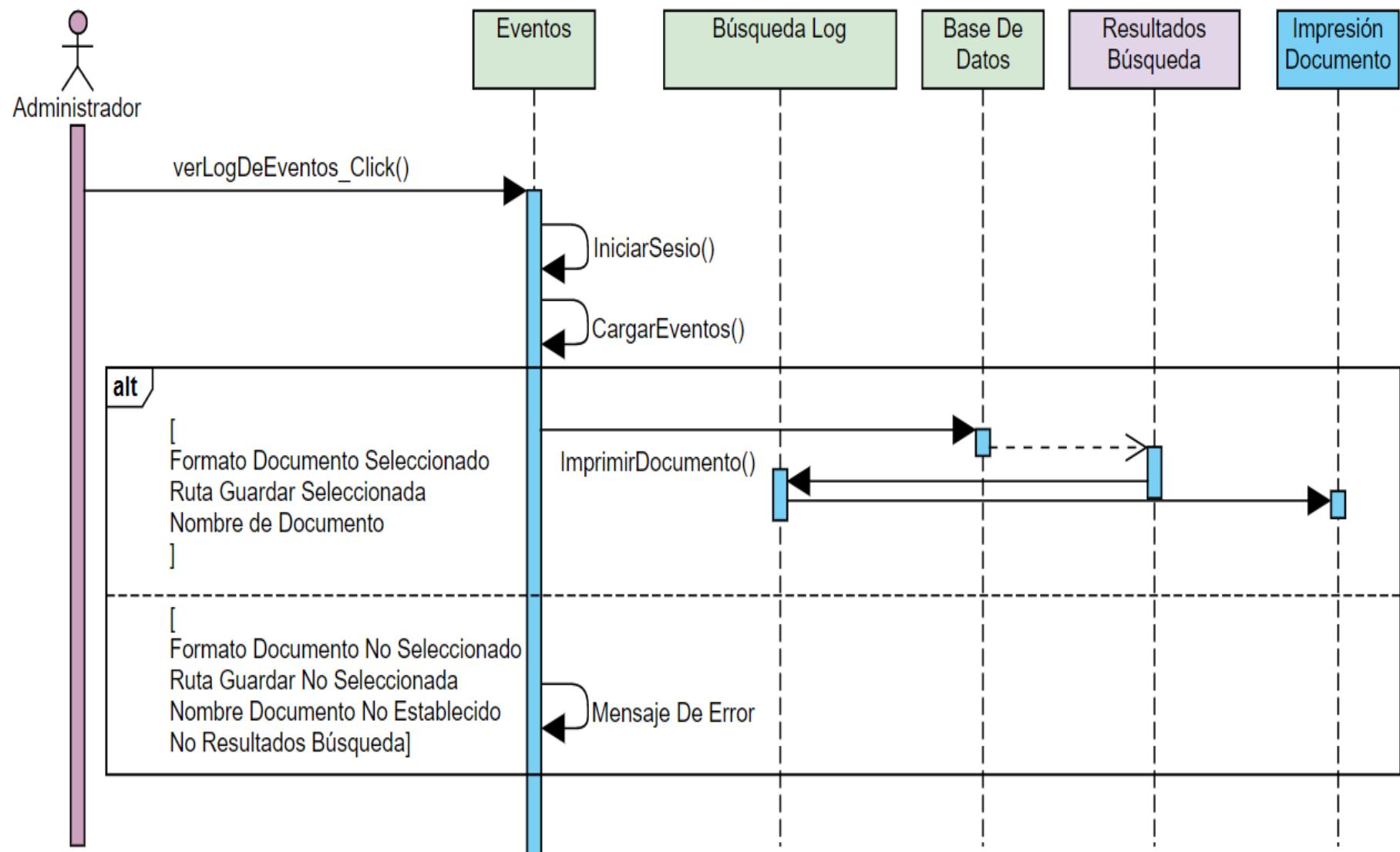


Figura 2.8. Diagrama de secuencia del log de eventos.

Los dispositivos de acceso como: cámaras, lectores dactilares y teclados numéricos deben ser activados o desactivados, según sea el caso, por el administrador en sus respectivos formularios. Para activar los dispositivos de acceso, éstos deben ser conectados y registrados; posteriormente, el administrador seleccionará la sala con la cual se desea establecer conexión con el dispositivo de acceso.

Finalmente, todos los dispositivos de acceso son desactivados en el formulario principal llamado “Sistema_Seguridad_FIEE”, seleccionando primero la sala de la cual desea desactivar el dispositivo.

Las imágenes obtenidas por las cámaras de video conectadas al sistema prototipo de seguridad pueden ser manipuladas cambiando el brillo, contraste gamma y saturación para poder obtener una mejor definición. Adicionalmente, se puede realizar una instantánea de la imagen; así, como un video de la cámara. Todas las modificaciones son realizadas en las propiedades de la cámara dentro del formulario principal “Sistema_Seguridad_FIEE”.

El caso de uso y el diagrama de secuencia para cuando el administrador realiza estas acciones se muestra en las Figuras 2.11 y 2.12 respectivamente.

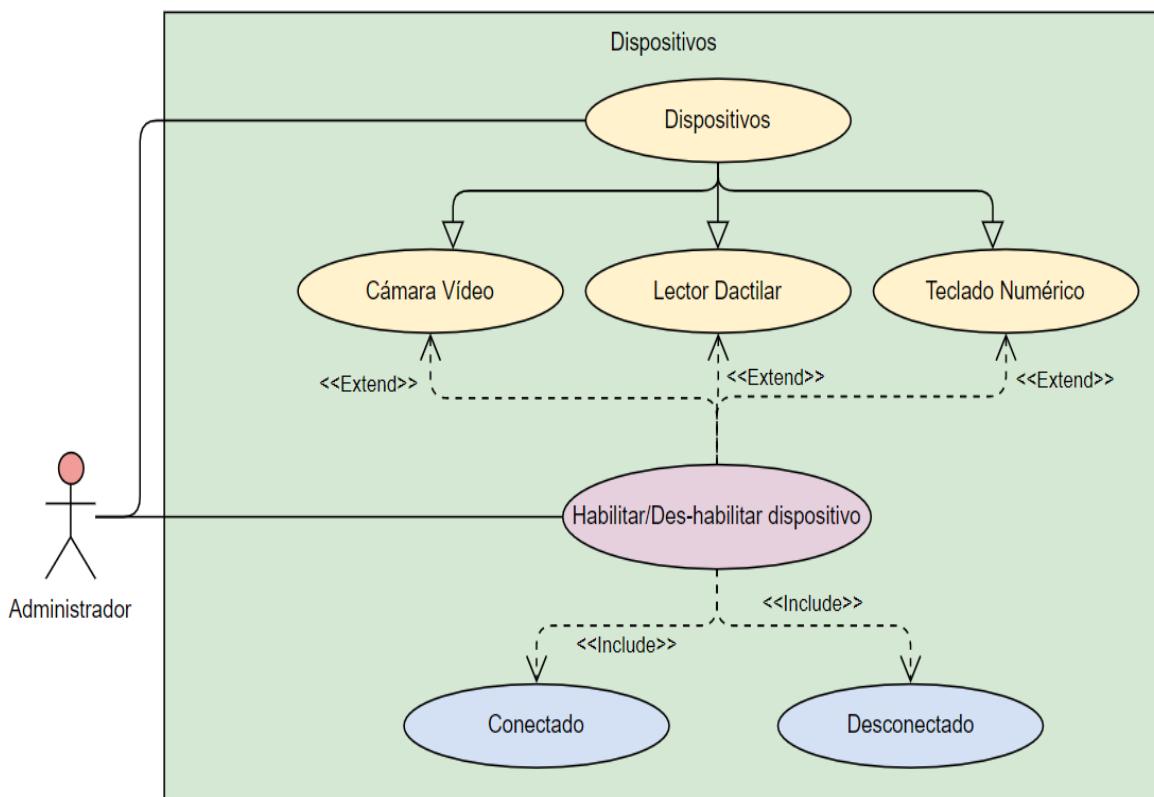


Figura 2.9. Diagrama de caso de uso para la administración de dispositivos.

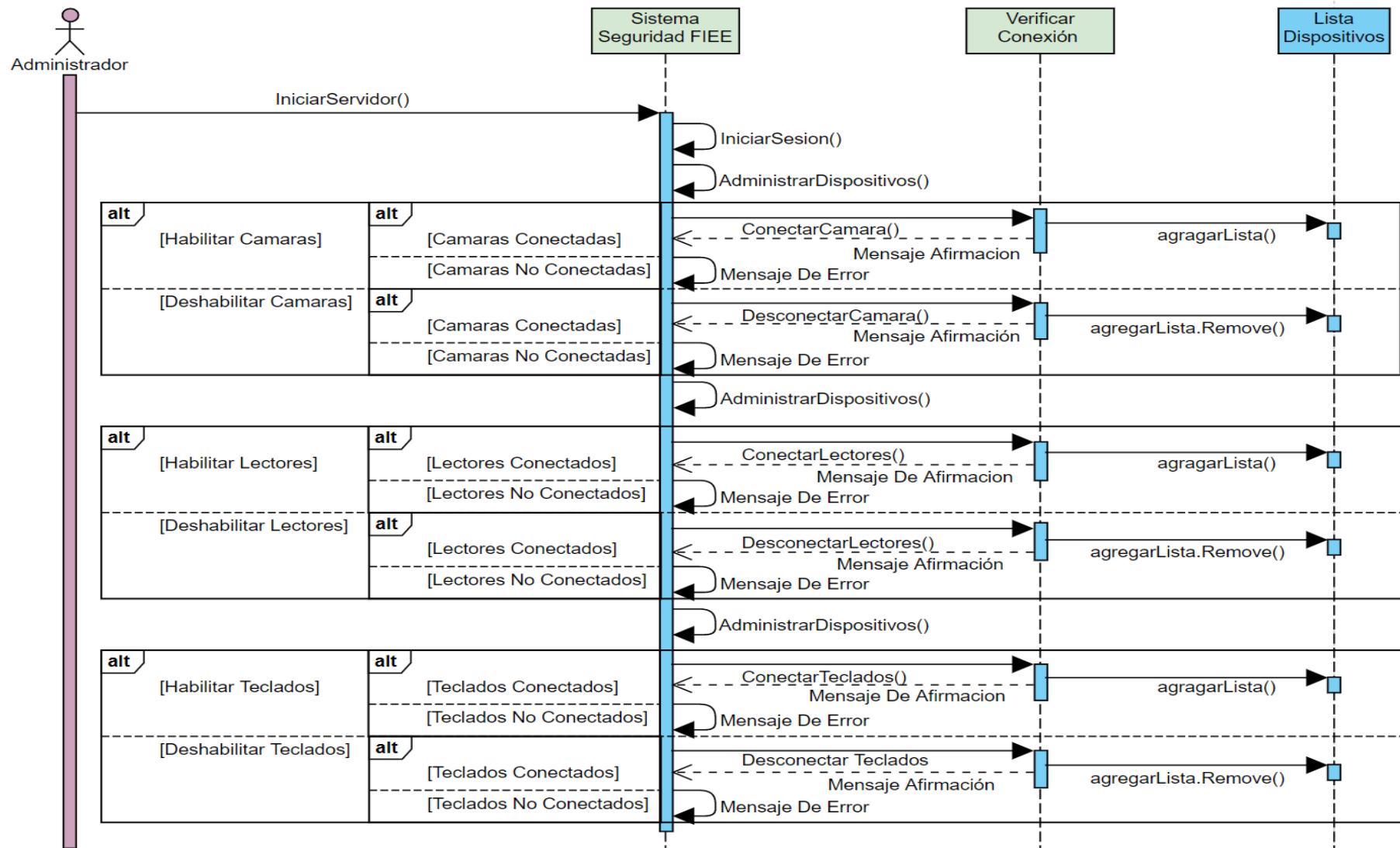


Figura 2.10. Diagrama de secuencia para la administración de dispositivos.

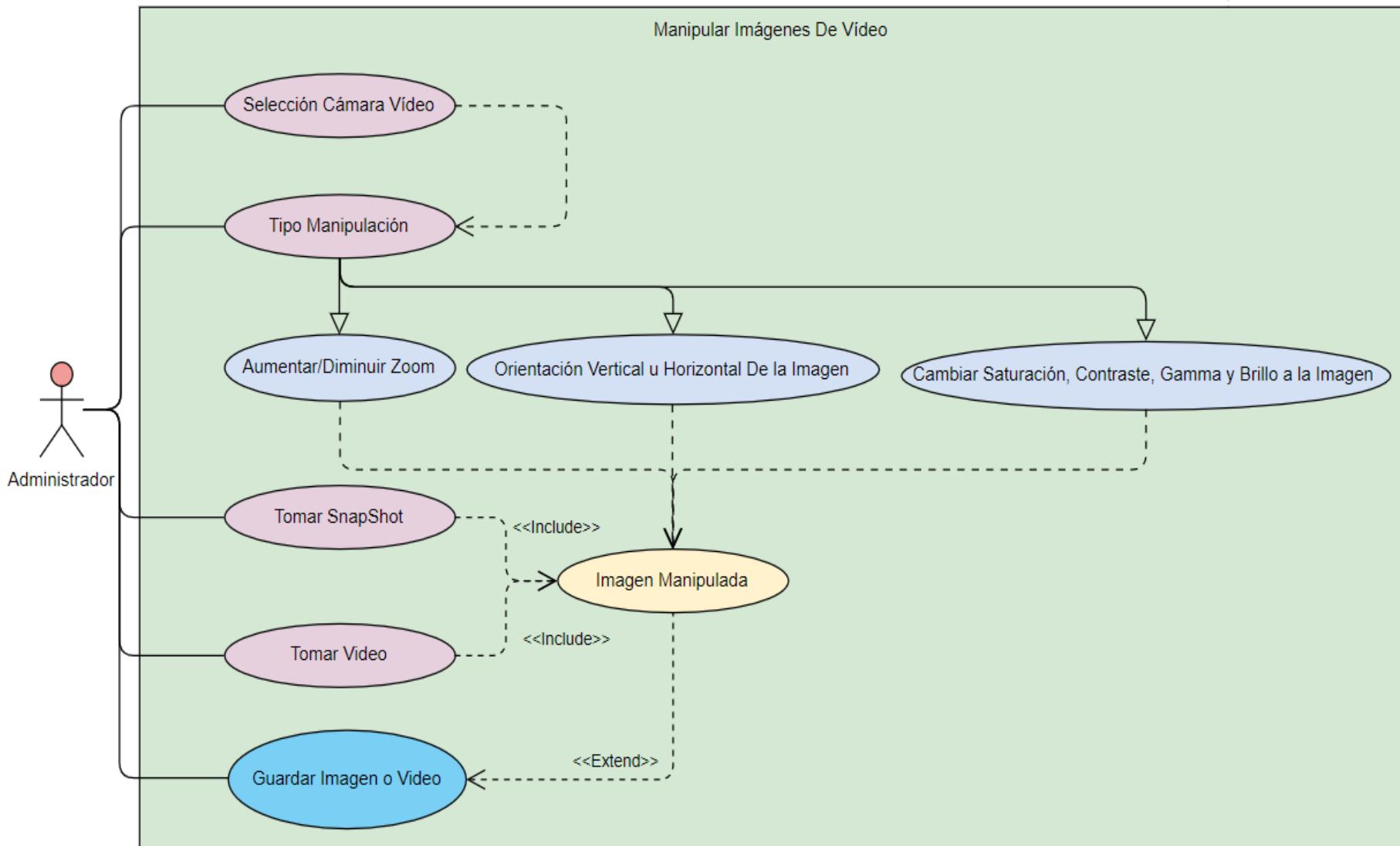


Figura 2.11. Diagrama de caso de uso para la manipulación de las imágenes de video.

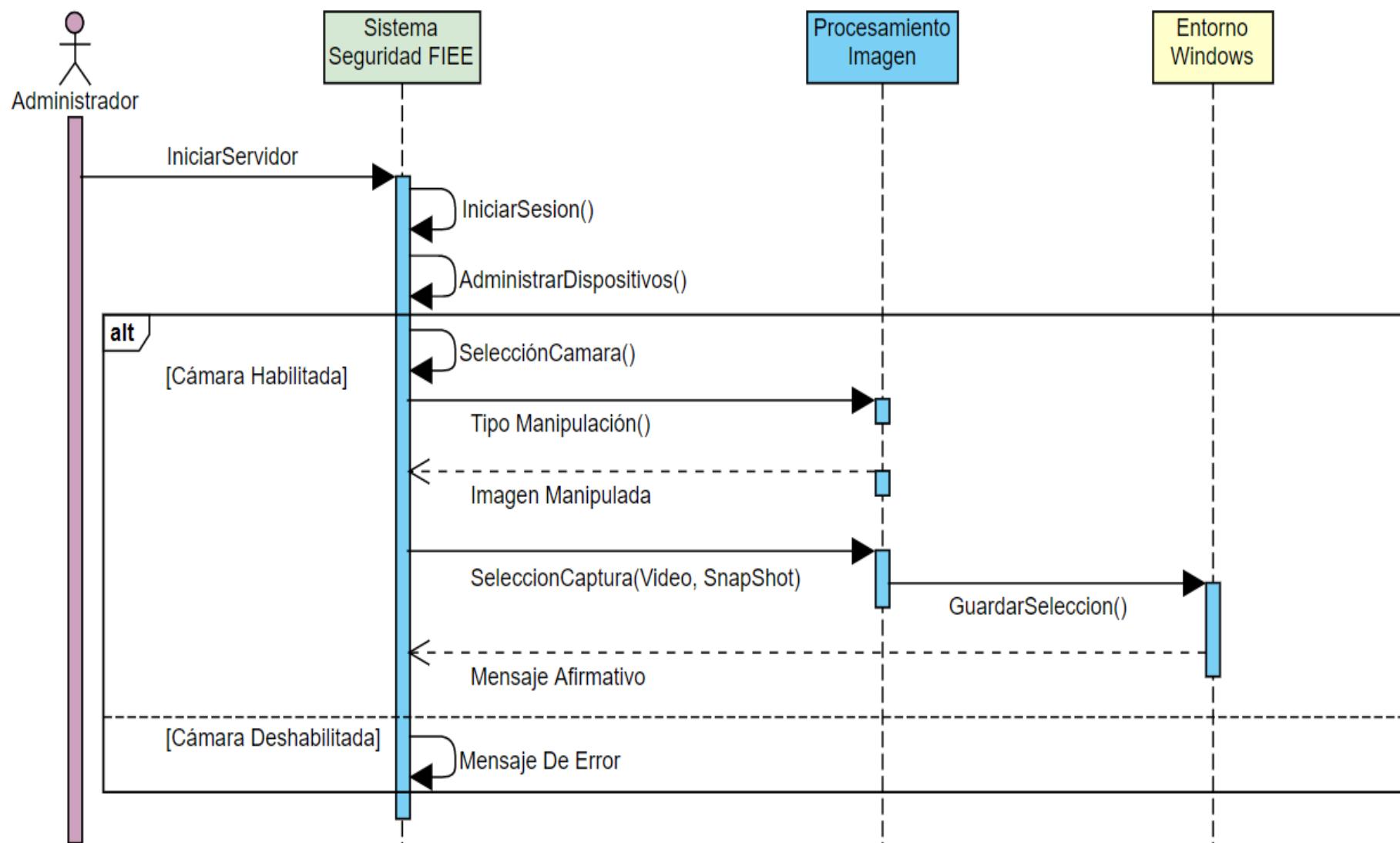


Figura 2.12. Diagrama de secuencia para la manipulación de las imágenes de video.

2.1.5. DIAGRAMAS DE CLASE DE LOS MÓDULOS: INTERFAZ DE USUARIO, LÓGICA DE NEGOCIOS Y ACCESO A DATOS

La aplicación de escritorio, del sistema prototipo de acceso a los laboratorios de redes de la FIEE de la EPN basado en reconocimiento facial, llamado “Sistema_Seguridad_FIEE” se encuentra dividido en dos proyectos. El primer proyecto llamado “Cliente” contiene la “Interfaz de Usuario” y la “Lógica de Negocios”; mientras que el segundo proyecto llamado “ObjetoRemoto” contiene al “Servidor” y el “Acceso a Datos”.

En la Figura 2.13 se puede observar de manera general todas las clases involucradas para los módulos “Interfaz de Usuario” y “Lógica de Negocios” incluyendo a las clases generadas por la plataforma como son la de “Program” y la de “Settings”. No se descarta las clases tipo “Forma” ya que dentro de ellas se encuentra parte de la “Lógica de Negocios”.



Figura 2.13. Diagrama de clases de la interfaz de usuario y la lógica de negocios.

Las diferentes clases involucradas en la “Interfaz de Usuario” no tienen relación entre si ya que estas solo representan los componentes gráficos que la componen y los métodos que utiliza para su funcionamiento. Se observa en la Figura 2.14 las diferentes clases para la “Interfaz de Usuario” con sus respectivos métodos. “Para más detalle ver el Anexo D”

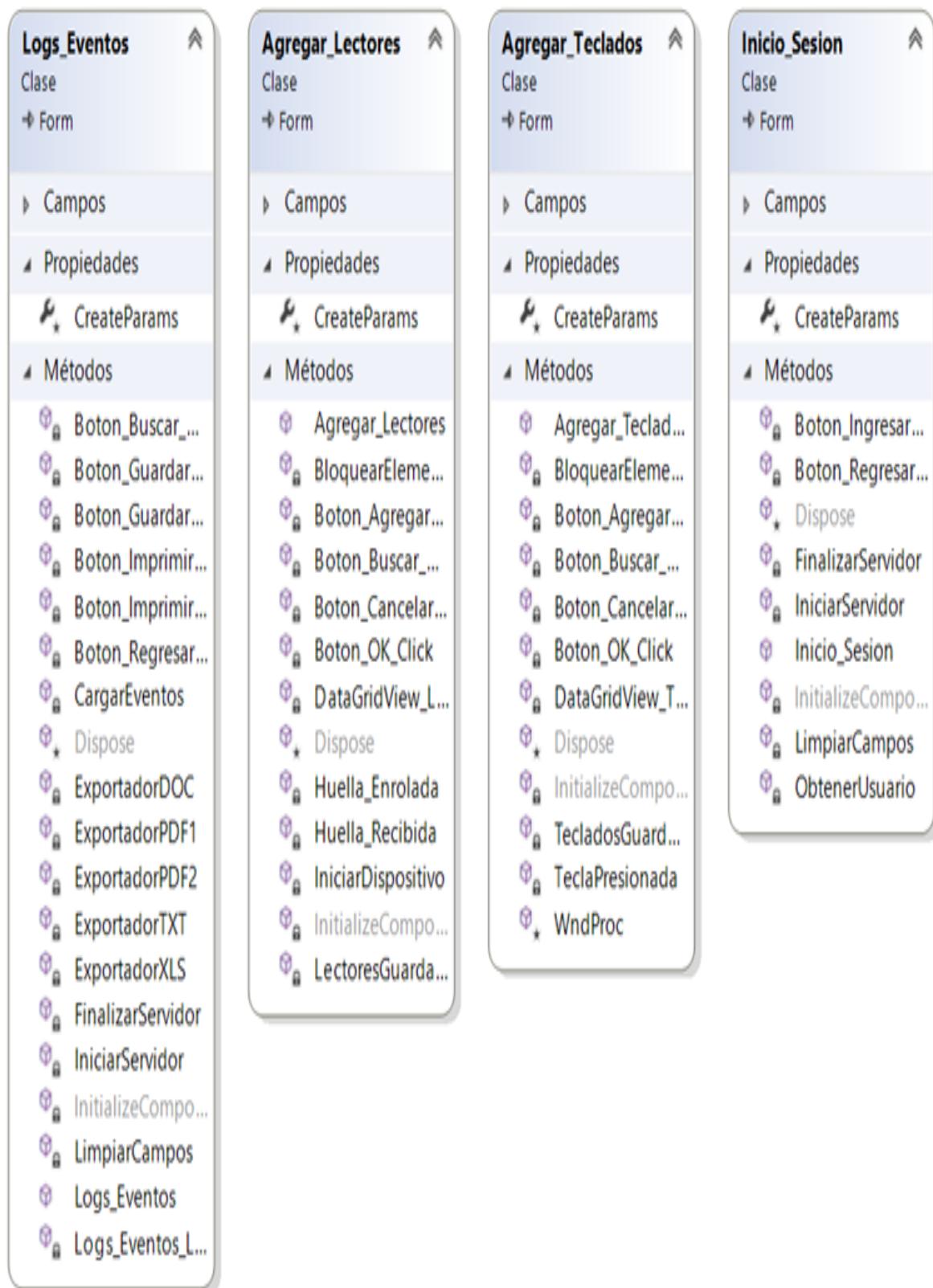


Figura 2.14. Diagrama de clases de la interfaz de usuario.

Para la Figura 2.15, se puede observar la “Lógica de Negocios” entre las diferentes clases del sistema prototipo.

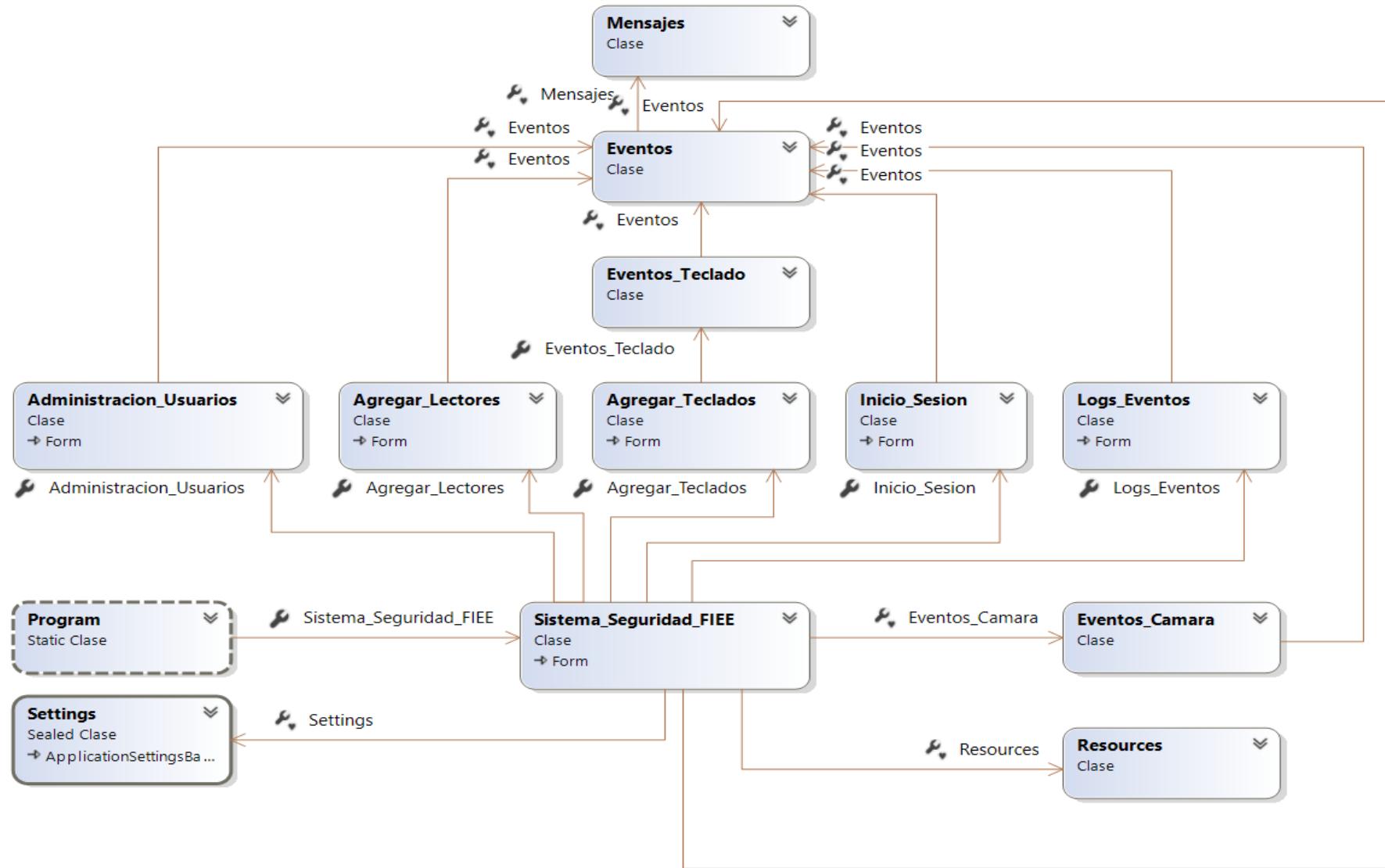


Figura 2.15. Diagrama de clases lógica de negocios.

En la Figura 2.16, se puede observar el diagrama de clase para el “Acceso a Datos”; para ello se inicializa el servidor, el cual se comunica con la clase “ObjetoRemoto” donde se encuentran las consultas LINQ to SQL; para realizar las consultas la clase “ObjetoRemoto” hace uso de las interfaces de “ObjetoRemoto” y al mismo tiempo se establece una comunicación con la base de datos para obtener resultados. En la base de datos se encuentran las tablas “Usuarios” y “Logs” en las cuales se realizan las consultas. Mediante el servidor y el ObjetoRemoto se devuelven las consultas al cliente.

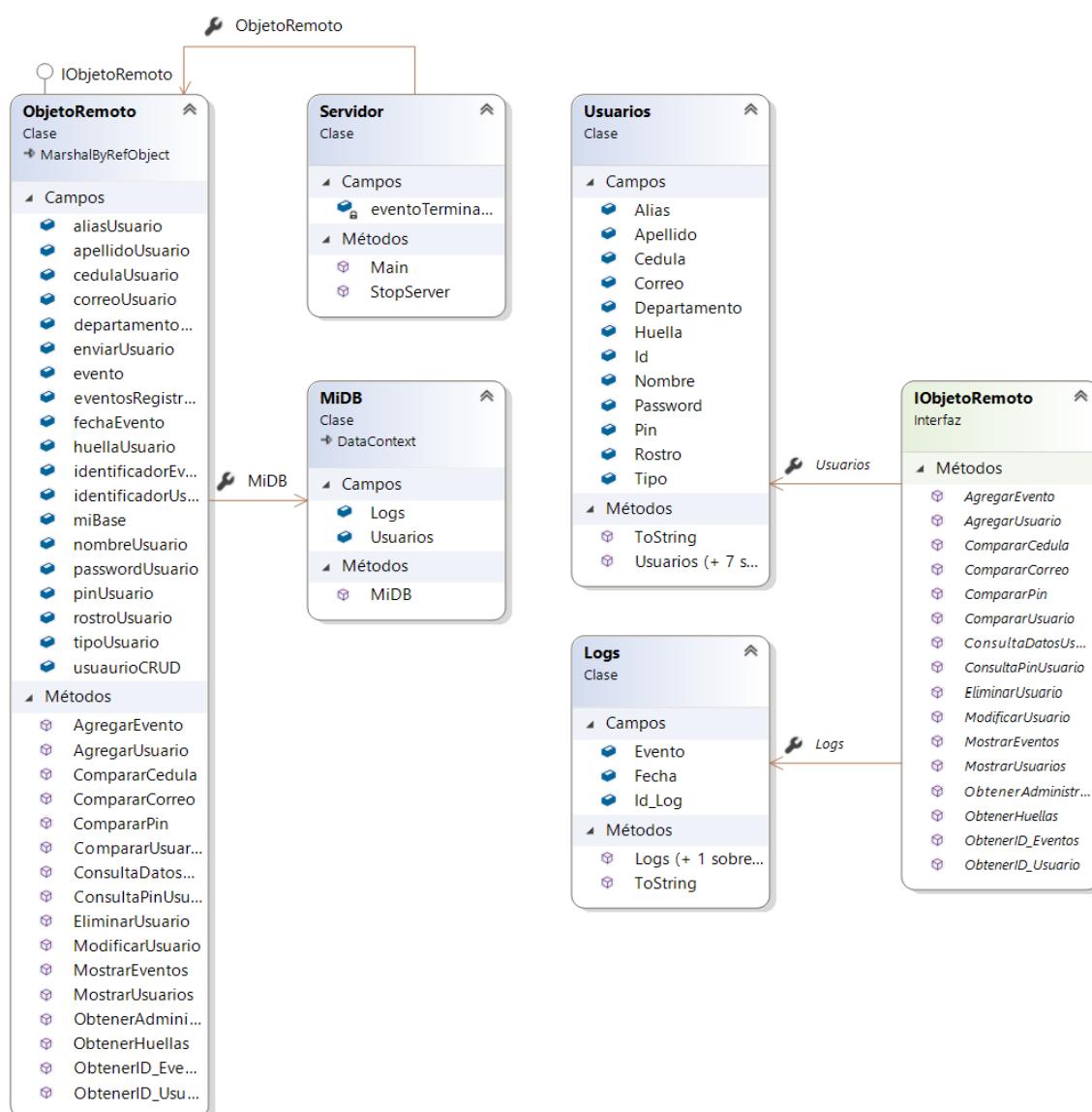


Figura 2.16. Diagrama de clases del acceso a datos y servidor.

Como se mencionó anteriormente el “Acceso a Datos” se encuentra dentro del proyecto “ObjetoRemoto” el mismo que contiene, adicionalmente, la base de datos y el servidor; en la Figura 2.17 se puede observar de manera general todas las clases involucradas.



Figura 2.17. Diagrama de clases del acceso a datos y servidor.

2.1.6. DIAGRAMAS DE ACTIVIDADES DE CADA MÓDULO: INTERFAZ DE USUARIO, LÓGICA DE NEGOCIOS Y ACCESO A DATOS

En los siguientes diagramas se muestra las actividades realizadas por el administrador o por un usuario al realizar una determinada acción; cabe mencionar que las actividades de “Lógica de Negocios” y “Acceso a Datos” se encuentran implícitas dentro de cada diagrama de la “Interfaz de Usuario”. Se debe recordar que las actividades que puede realizar el administrador son: login, administrar usuarios y ver e imprimir eventos y para el usuario general es la de proporcionar los datos para el registro o modificación al administrador y la de usar los dispositivos de acceso.

En la Figura 2.18 se puede apreciar el diagrama de actividades para el login del administrador. El diagrama de actividades para el Registro, Actualización y Eliminación de usuarios se muestra en la Figura 2.19. La visualización, filtrado e impresión de los eventos generados durante el uso de la aplicación se muestra la Figura 2.20.

El administrador es capaz de habilitar o deshabilitar los dispositivos periféricos, el diagrama de actividades que describe el procedimiento se muestra en la Figura 2.21. De igual manera el administrador es capaz de modificar las imágenes obtenidas por la cámara de video y si lo desea puede almacenarlas en el ordenador donde se está ejecutando la aplicación, el diagrama de actividades se lo muestra en la Figura 2.22.

La “Lógica de Negocios” y el “Acceso a Datos” se muestra en el mismo diagrama, ya que la “Lógica de Negocios” es la que se encarga de comunicar al cliente con el servidor y éste a su vez se comunica con la base de datos; la “Lógica de Negocios” es la cual permite, por ejemplo, agregar o quitar dispositivos periféricos. El diagrama de actividades para la “Lógica De Negocios” y “Acceso A Datos” se muestra en la Figura 2.23.

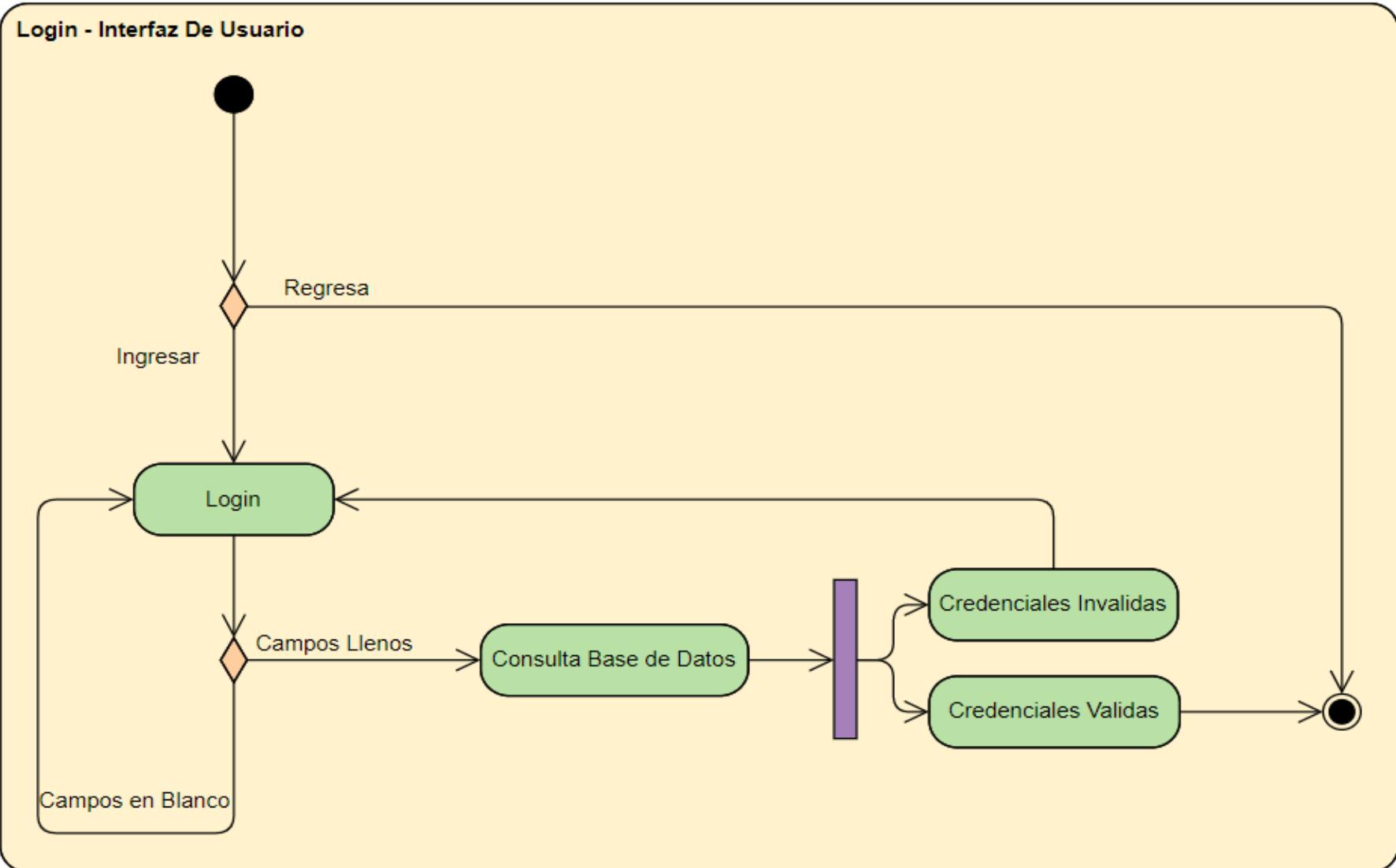


Figura 2.18. Diagrama de actividades del login.

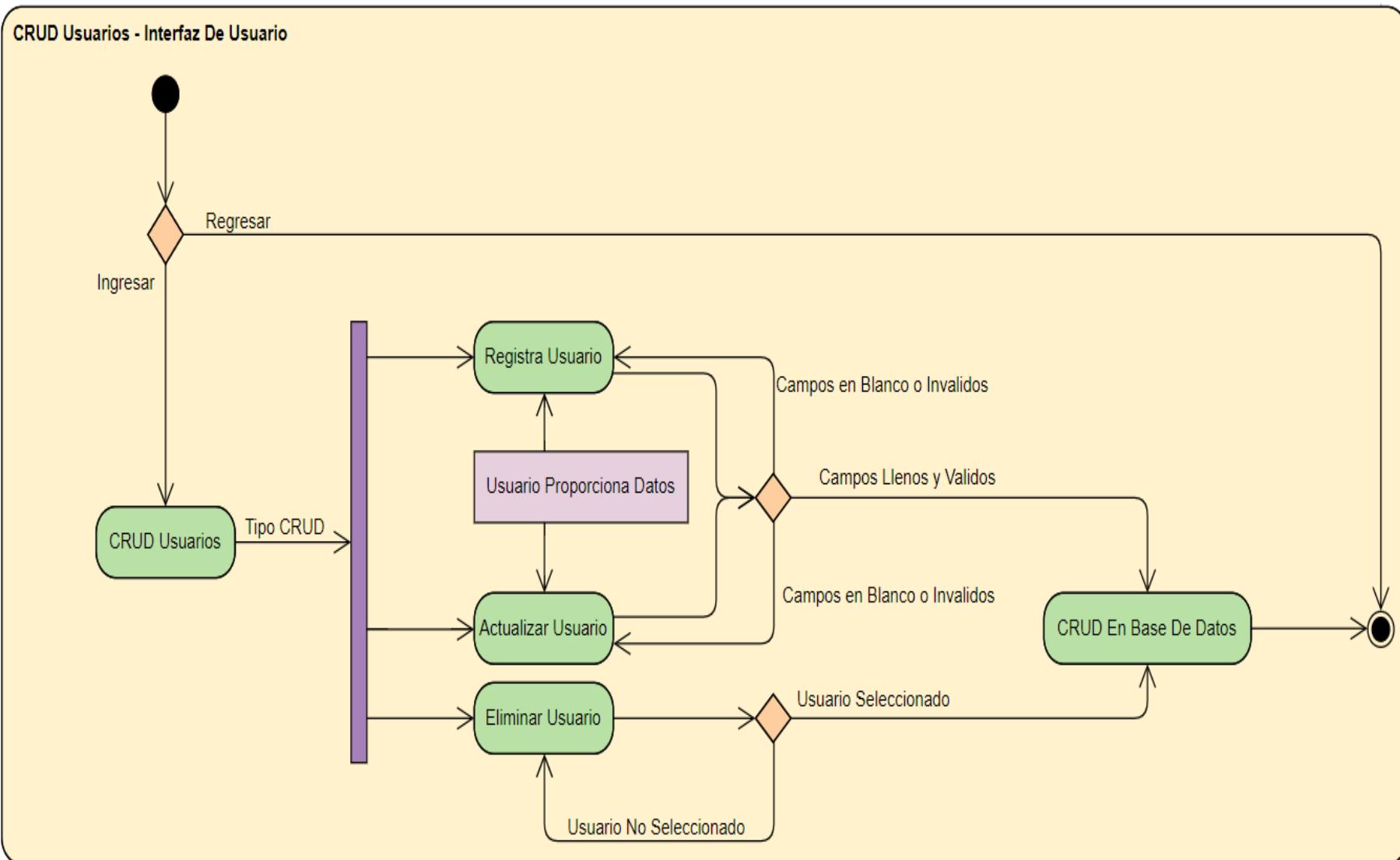


Figura 2.19. Diagrama de actividades del CRUD usuarios.

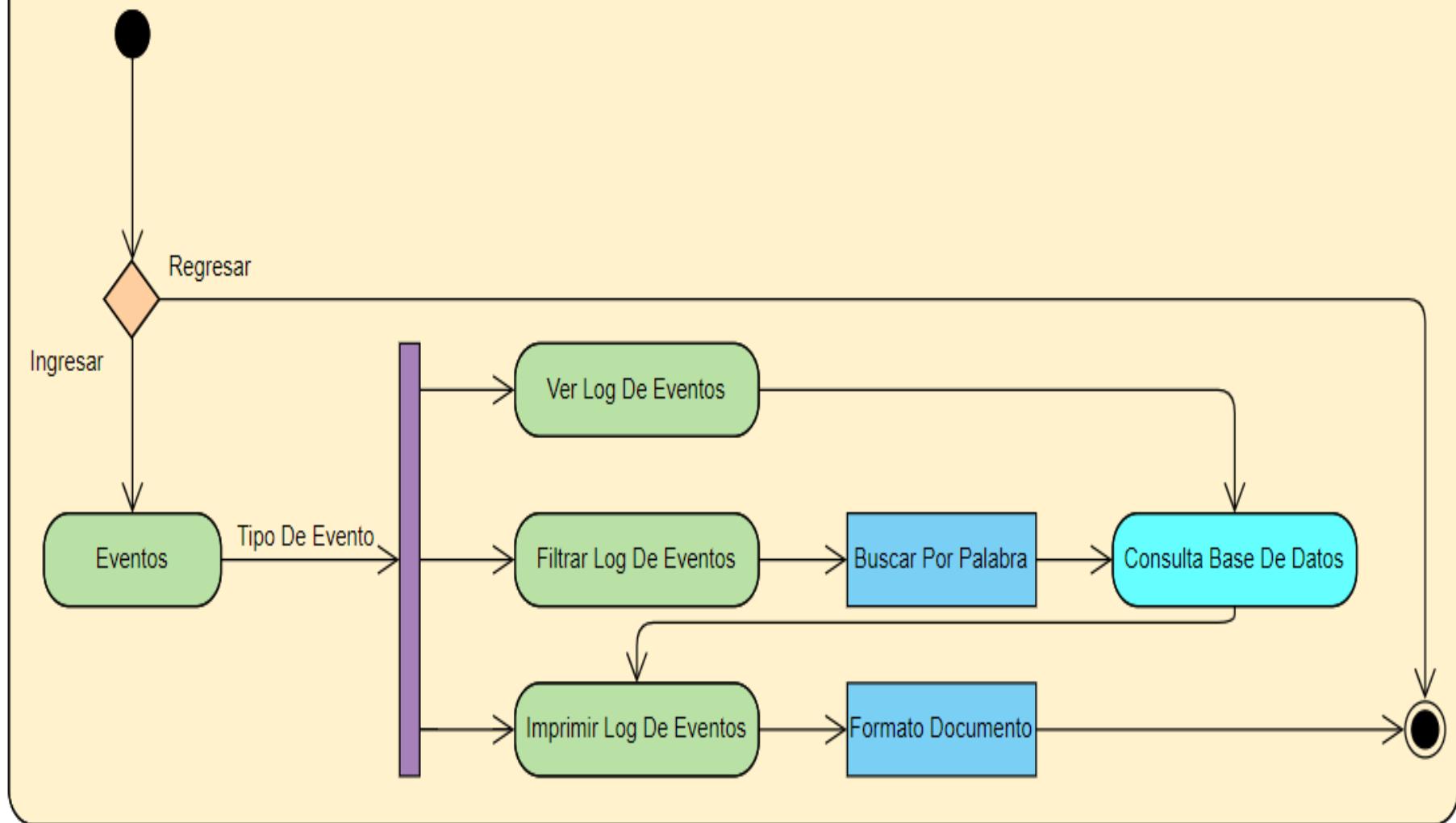
Eventos - Interfaz De Usuario

Figura 2.20. Diagrama de actividades del log de eventos.

Dispositivos -Interfaz De Usuario

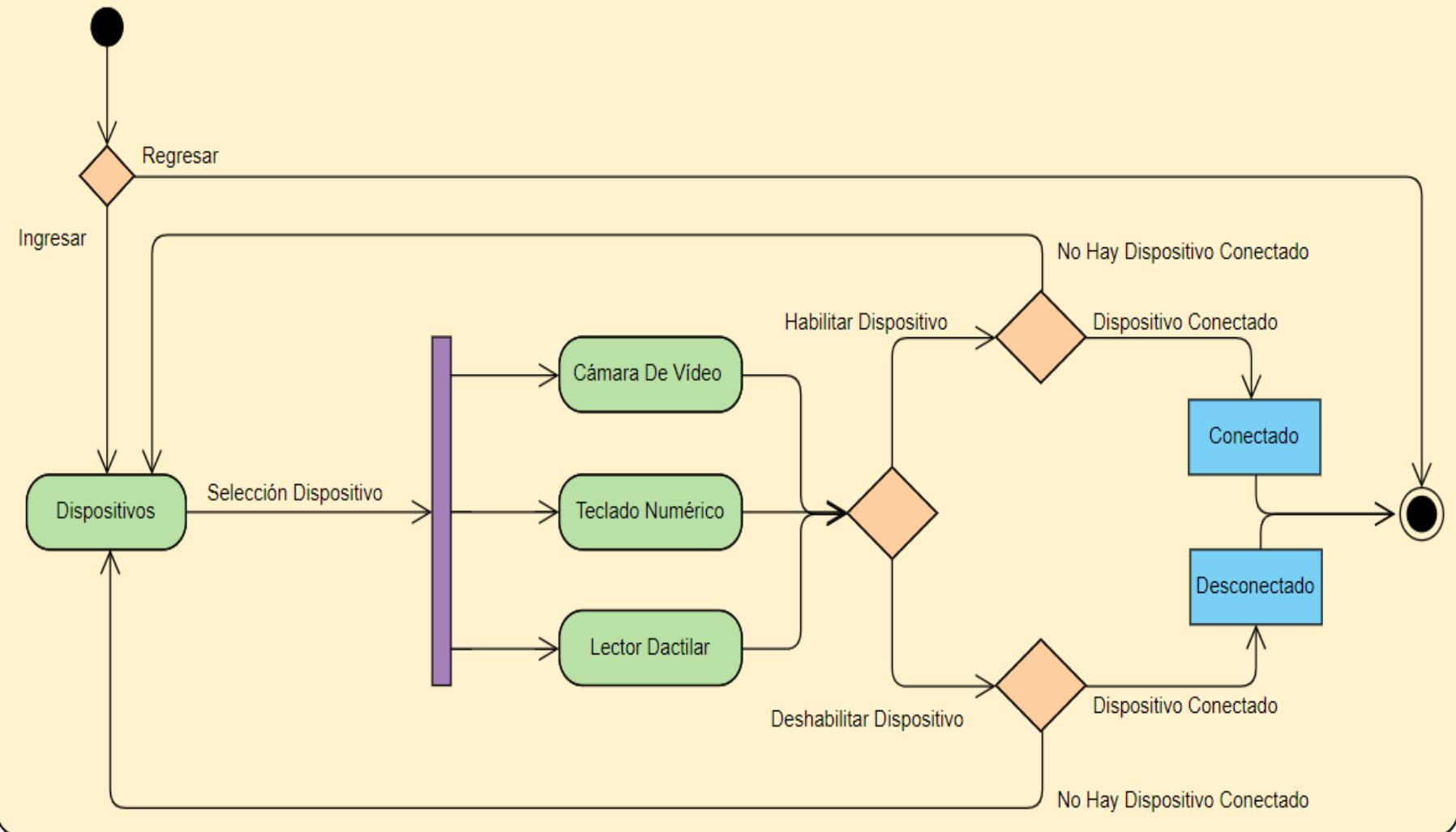


Figura 2.21. Diagrama de actividades de la administración de dispositivos.

Manipular Imágenes de Vídeo

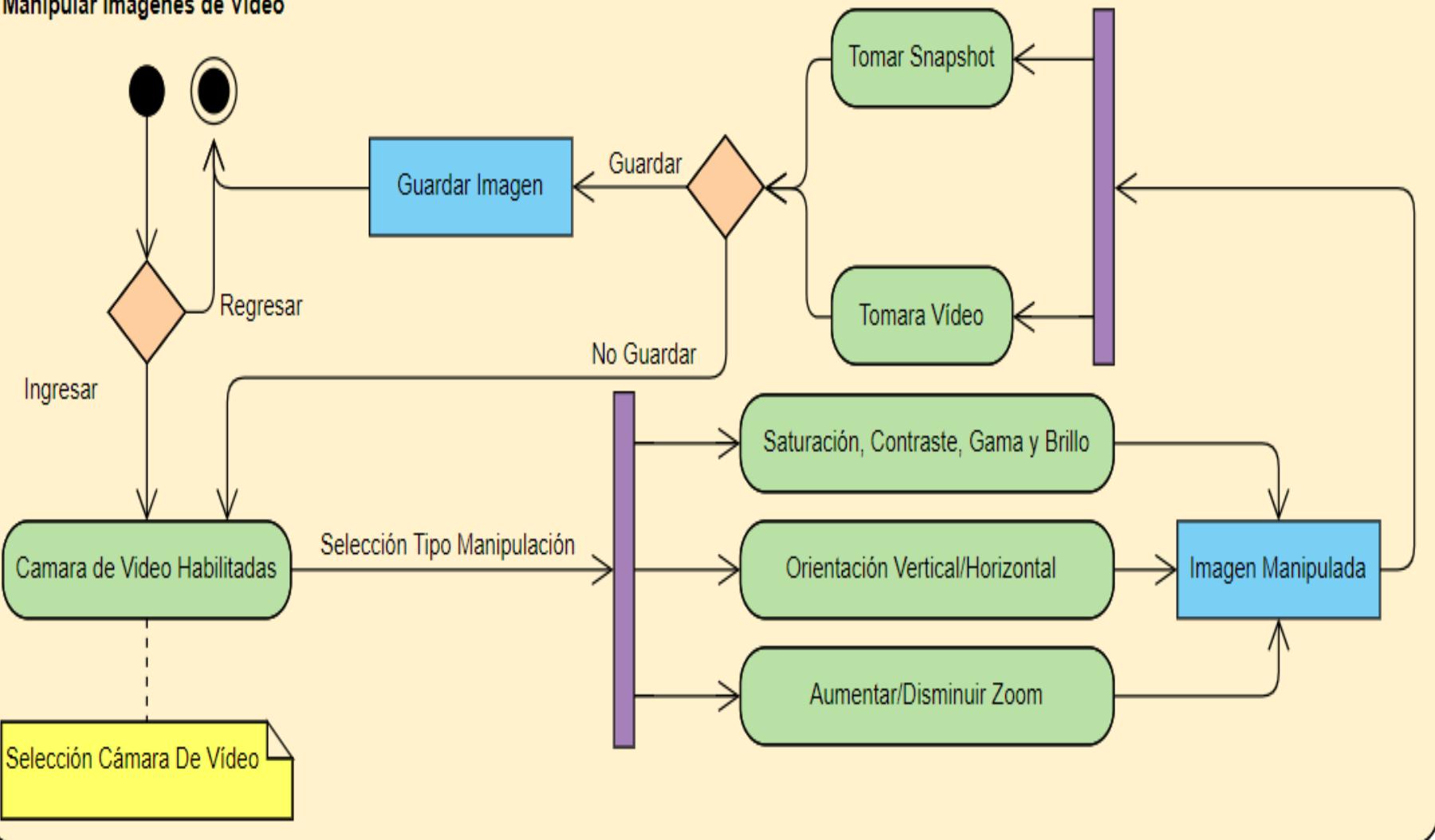


Figura 2.22. Diagrama de actividades de la manipulación de las imágenes de Video.

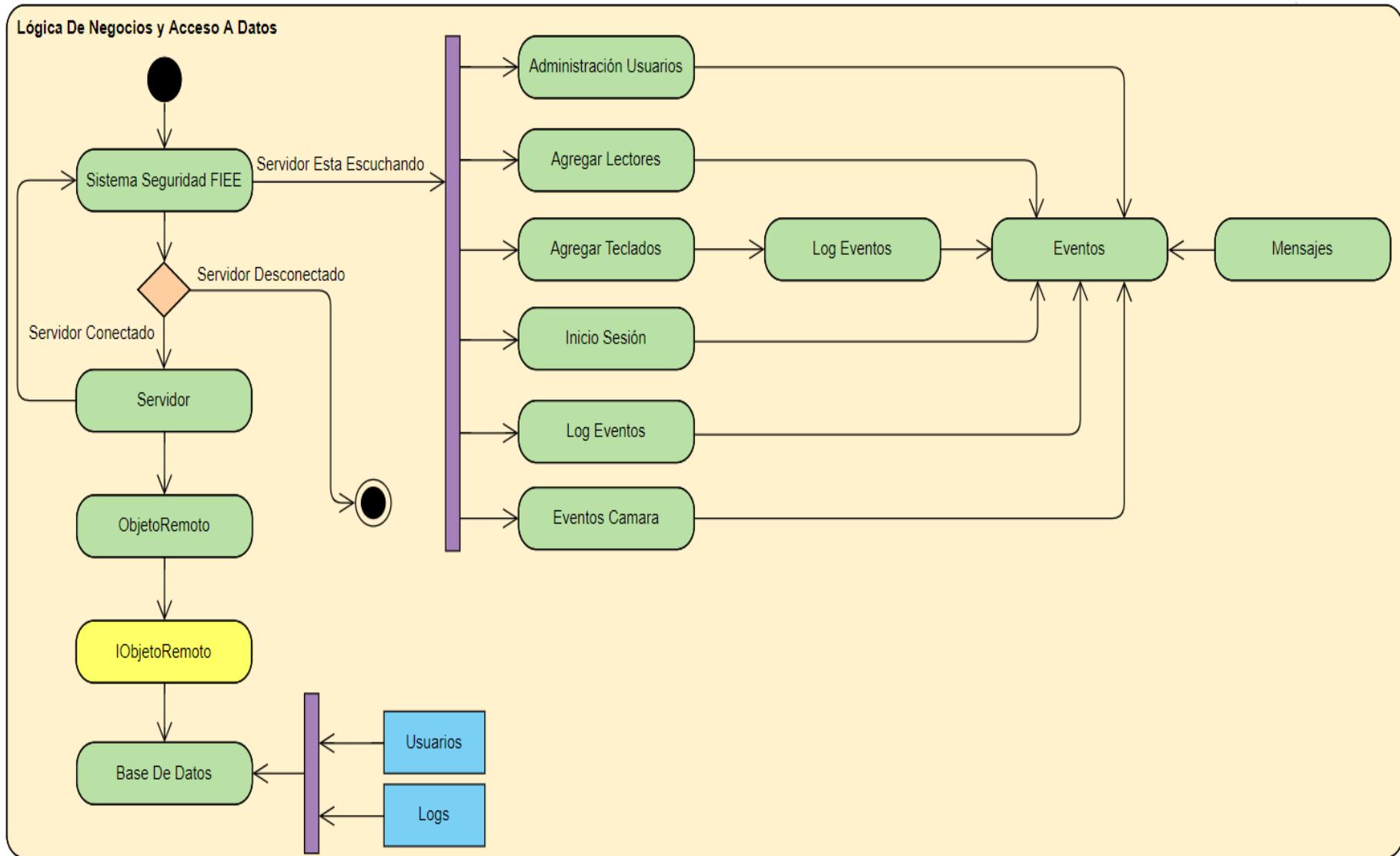


Figura 2.23. Diagrama de actividades lógica de negocios y acceso a datos.

2.1.7. DIAGRAMAS ENTIDAD RELACIÓN DEL MÓDULO BASE DE DATOS

Un diagrama entidad relación ilustra cómo las entidades, como personas, se relacionan entre sí dentro de un sistema. Para el diseño del sistema prototipo se hizo uso de dos tablas; la primera tabla llamada “Usuarios” es utilizada para registrar a los usuarios y la segunda tabla llamada “Logs” es usada para registrar los eventos ocurridos al usar la aplicación de escritorio como: registrar o eliminar usuarios. El diagrama entidad relación de la base de datos se muestra en la Figura 2.24.

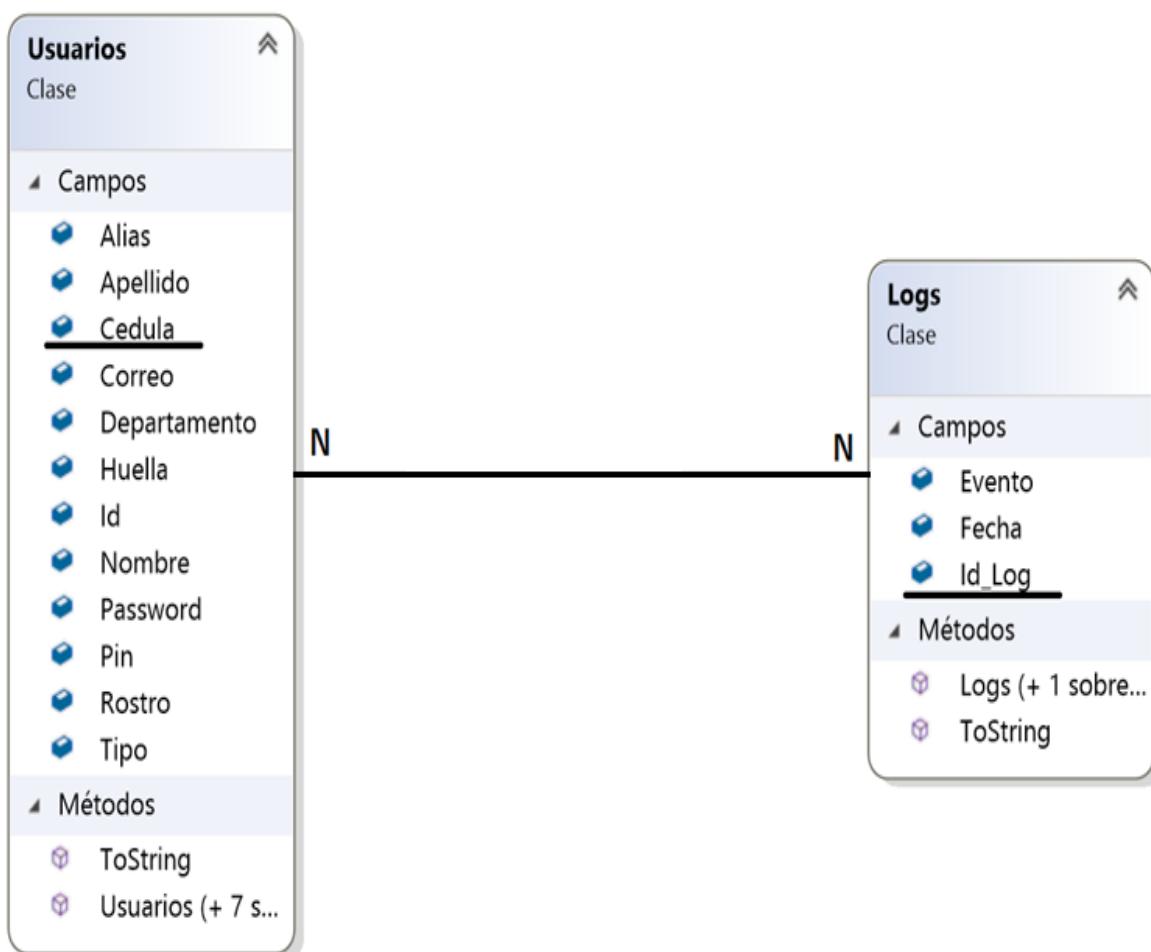
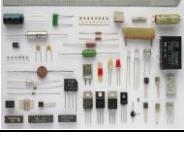


Figura 2.24. Diagrama entidad relación del módulo base de datos.

2.1.8. REQUERIMIENTOS FÍSICOS DEL SISTEMA DE SEGURIDAD

Para la implementación del sistema de seguridad prototipo se requieren de varios elementos físicos; en la Tabla 2.5 se muestran los elementos físicos utilizados para la implementación del sistema de seguridad, detallando la descripción, cantidad y una imagen de referencia.

Tabla 2.5. Requerimientos físicos del sistema de seguridad.

REQUERIMIENTOS FÍSICOS DEL SISTEMA DE SEGURIDAD		
ELEMENTO	DESCRIPCIÓN	IMAGEN
Computador	Un computador con sistema operativo Windows 10.	
Cámara IP	Cámaras IP “Intelligent Camera” Onvif YY HD WIFI Audio. Códice H.264, Puerto Ethernet, Protocolos: HTTP, TCP, UDP, etc.	
Lector dactilar	Lectores dactilares ZK9500 Fingerprint Reader Bio4. Sensor óptico, Resolución 500dpi, puerto USB 2.0.	
Panel numérico	Teclados numéricos ThinkTec KBF17, puerto USB.	
Cable UTP con interfaz física RJ45	El Cable UTP debe ser de la categoría 5e con la norma EIA/TIA-568-B.	
Router	Router Cable/DSL TL-R460 de 4 puertos LAN 10/100Mbps y 1 puerto WAN.	
USB Hub	USB Hub de 4 puertos 2.0 y 480 Mbps de velocidad.	
Arduino	Arduino Uno con microcontrolador ATmega328 y voltaje operativo de 5v DC.	
Cerradura Magnética	Cerradura magnética AL-180 ZKTeco y voltaje operativo de 12v DC. Fuerza de retención: 150 Kg (equivalentes a 330 libras).	
Otros	Cables conectores para Arduino, extensiones eléctricas, transformadores de 12v DC, leds, relés, resistencias y canaletas.	

En la Tabla 2.6 se muestra la segunda actualización del tablero Kanban, en ella se muestra que las tareas que se encontraban en proceso han finalizado y nuevas tareas ocupan su posición. Para este punto, las tareas que se encuentran en proceso corresponden a la fase de implementación del sistema prototipo de seguridad.

Tabla 2.6. Segunda Actualización Tablero Kanban.

TABLERO KANBAN		
TAREAS POR REALIZAR	TAREAS EN PROCESO	TAREAS FINALIZADAS
Comprobaciones de funcionamiento de los módulos	Instalación Visual Studio 2017.	Selección metodología de Desarrollo Software.
Correcciones de los módulos.	Instalación Librerías Ozeki, EmguCV, OpenCV, ZKTeco e iText.	Ánalisis situación actual laboratorios de la FIEE.
Verificar la existencia de errores producidos al interconectar los equipos.	Creación de los formularios de la interfaz gráfica.	Encuesta de requerimientos.
Depurar los errores producidos al interconectar los equipos.	Implementación de cada módulo.	Diagramas de casos de uso.
Probar el funcionamiento de las cámaras, lectores de huellas y pines de acceso numérico.	Configuración de los terminales de acceso.	Diagramas de secuencia.
Se realizarán las encuestas de satisfacción.	Configuraciones de equipos intermedios.	Diagramas de actividades.
	Ubicación física de los terminales de acceso.	Diagramas de clase.
	Extensión del cableado necesario para la comunicación entre los equipos.	Diagramas entidad relación.
		Requerimientos físicos del sistema de seguridad.

2.2. IMPLEMENTACIÓN DEL SISTEMA PROTOTIPO

A continuación, se muestran los pasos realizados para la implementación de este proyecto:

- Instalación Visual Studio 2017.
- Instalación Librerías Ozeki, EmguCV, OpenCV, ZKTeco e iText.
- Creación de los formularios de la interfaz gráfica.
- Implementación de cada módulo.
- Configuración de los terminales de acceso
- Configuraciones de equipos intermedios.

- Ubicación física de los terminales de acceso.
- Extensión del cableado necesario para la comunicación entre los equipos.

2.2.1. INSTALACIÓN DE VISUAL STUDIO 2017

El primer paso para el desarrollo del sistema prototipo es la instalación de la plataforma Visual Studio 2017, la cual permitirá realizar la programación de cada uno de los módulos del sistema prototipo. Para poder obtener el instalador se puede ir a la página oficial de Microsoft (<https://visualstudio.microsoft.com/es/vs/>) y descargar el instalador; para este caso particular la Escuela Politécnica Nacional ofrece a su personal la opción de descargar Visual Studio con licenciamiento de la página de servicios institucionales (<https://servicios-it.epn.edu.ec/index.php/descargas>), como se observa en la Figura 2.25.

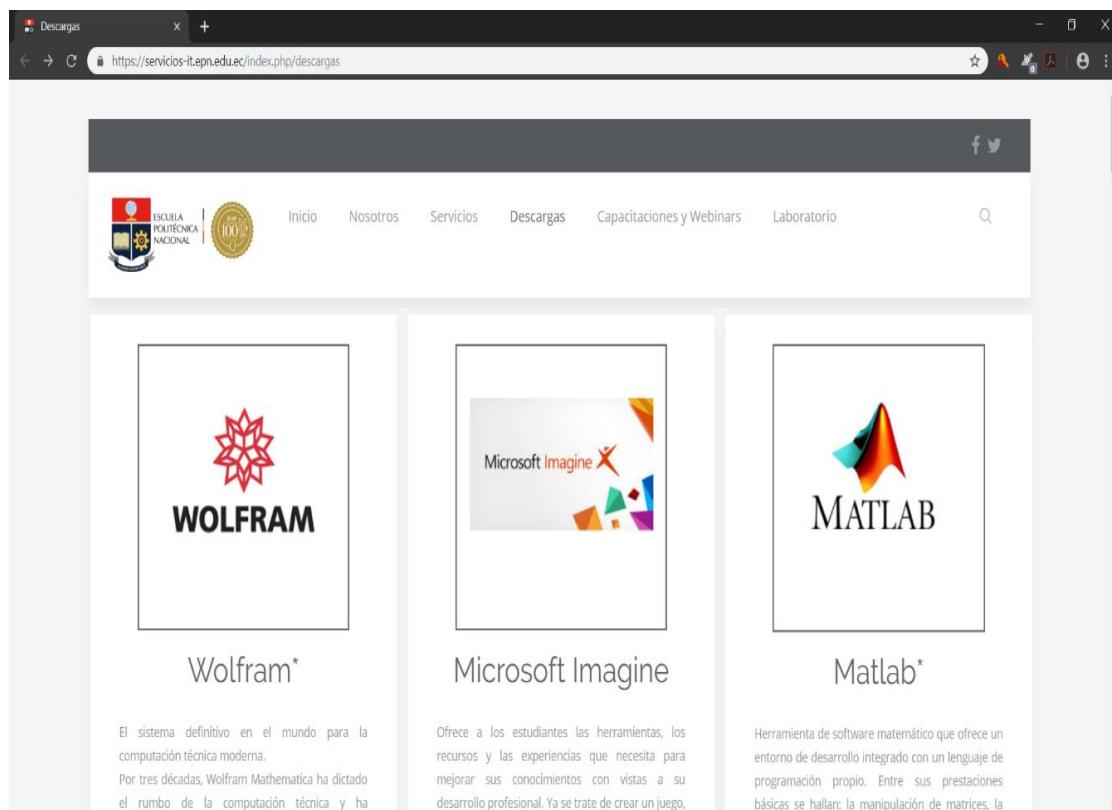


Figura 2.25. Página para descarga de Visual Studio.

Una vez que se haya ingresado todas las credenciales, se descargará automáticamente el instalador de Visual Studio; a éste se lo debe ejecutar como administrador para proceder con la instalación propia de Visual Studio; dentro del instalador se escogerá las opciones que más se ajusten a nuestras necesidades y se las instalará, esto se puede ver en la Figura 2.26.

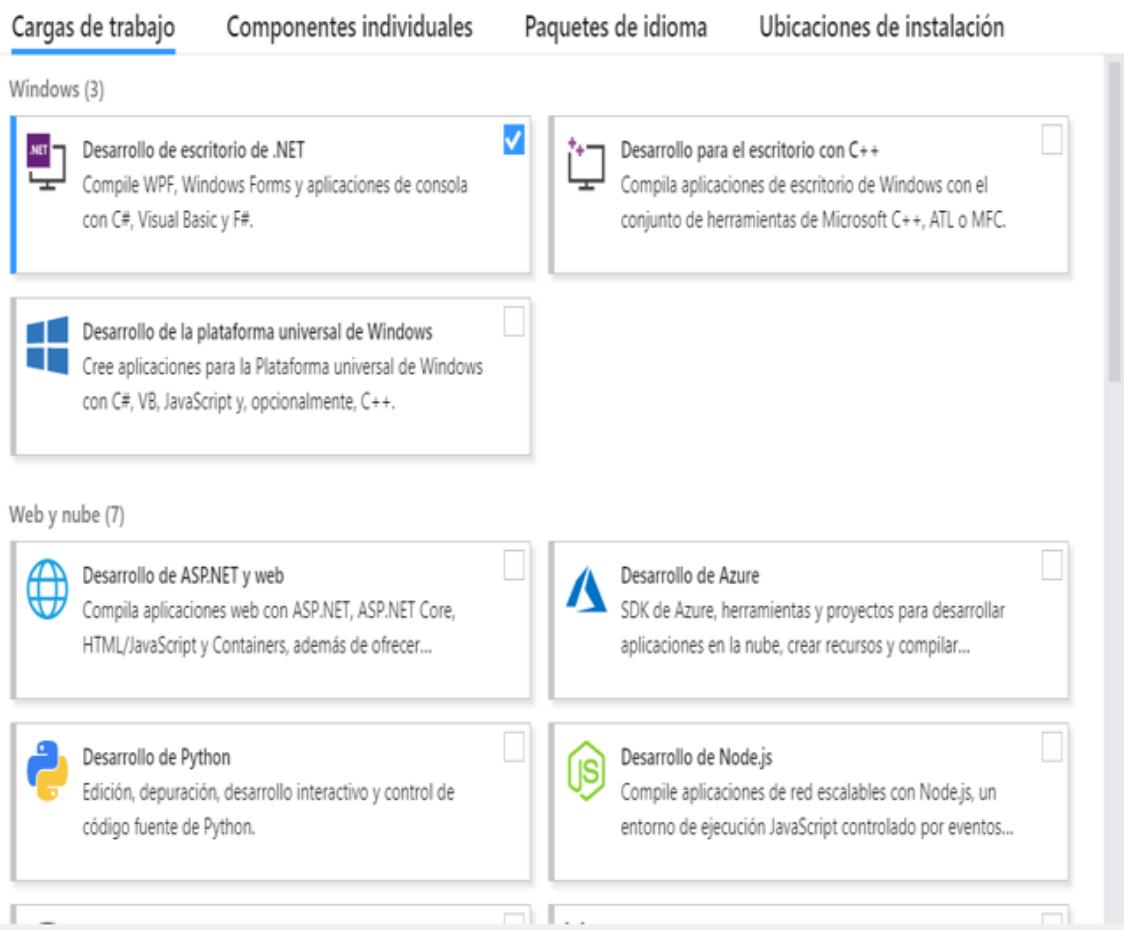


Figura 2.26. Parámetros posibles a instalar en Visual Studio.

Cuando haya terminado la instalación de Visual Studio, el mismo se iniciará automáticamente y se deberá proceder con el registro para poder usarlo a su máximo potencial y sin ningún tipo de restricciones. En la Figura 2.27 se puede observar la pantalla de inicio de Visual Studio una vez instalado.

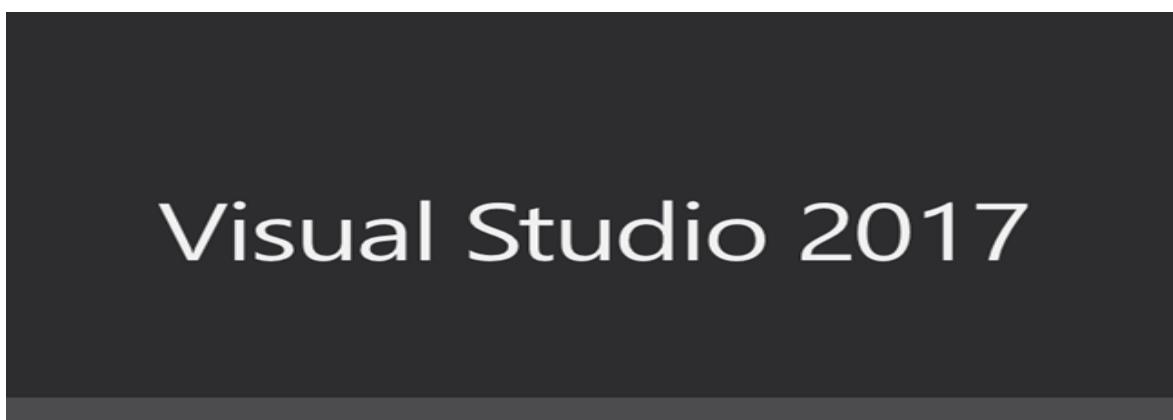


Figura 2.27. Pantalla de inicio de Visual Studio.

2.2.2. INSTALACIÓN LIBRERÍAS OZEKI, EMGUCV, OPENCV, ZKTECO E ITEXT

Antes de la programación como tal corresponde instalar las librerías necesarias. Para ello hay que descargar las librerías de OpenCV (<https://opencv.org/opencv-4-0-0.html>) y EmguCV (<https://sourceforge.net/projects/emgucv/>) de sus respectivas páginas, como se puede observar en las Figuras 2.28 y la 2.29.



Figura 2.28. Descarga librería OpenCV.

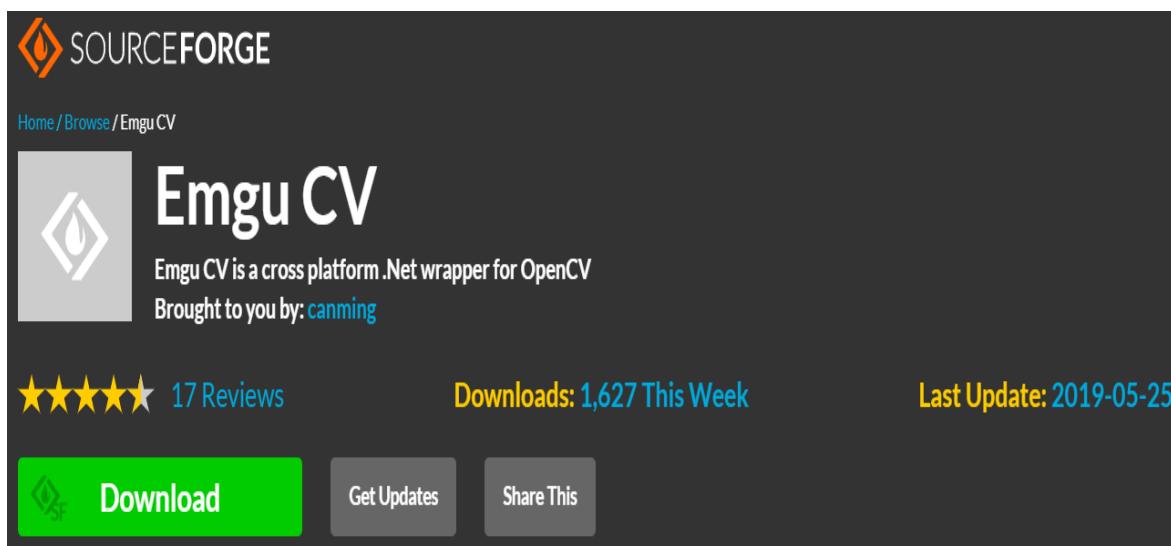


Figura 2.29. Descarga librería EmguCV.

Una vez descargadas las librerías se procede a la instalación de las mismas. Primero se instalará la librería de OpenCV; se ejecuta el archivo descargado como administrador y se busca donde se quiere extraer los archivos, ver Figura 2.30, y se espera mientras se extraen los archivos en la ruta especificada.

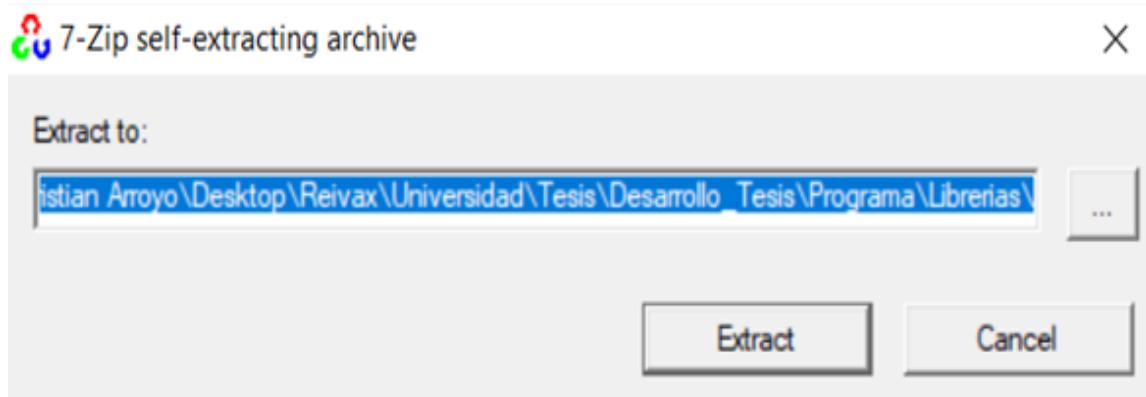


Figura 2.30. Ruta de extracción archivos OpenCV.

Al finalizar se puede observar que se ha creado una nueva carpeta en la ubicación seleccionada, ver Figura 2.31. En esta carpeta se encuentran las librerías a usar en la programación del proyecto; así como código útil que se puede usar como referencia al momento de programar.

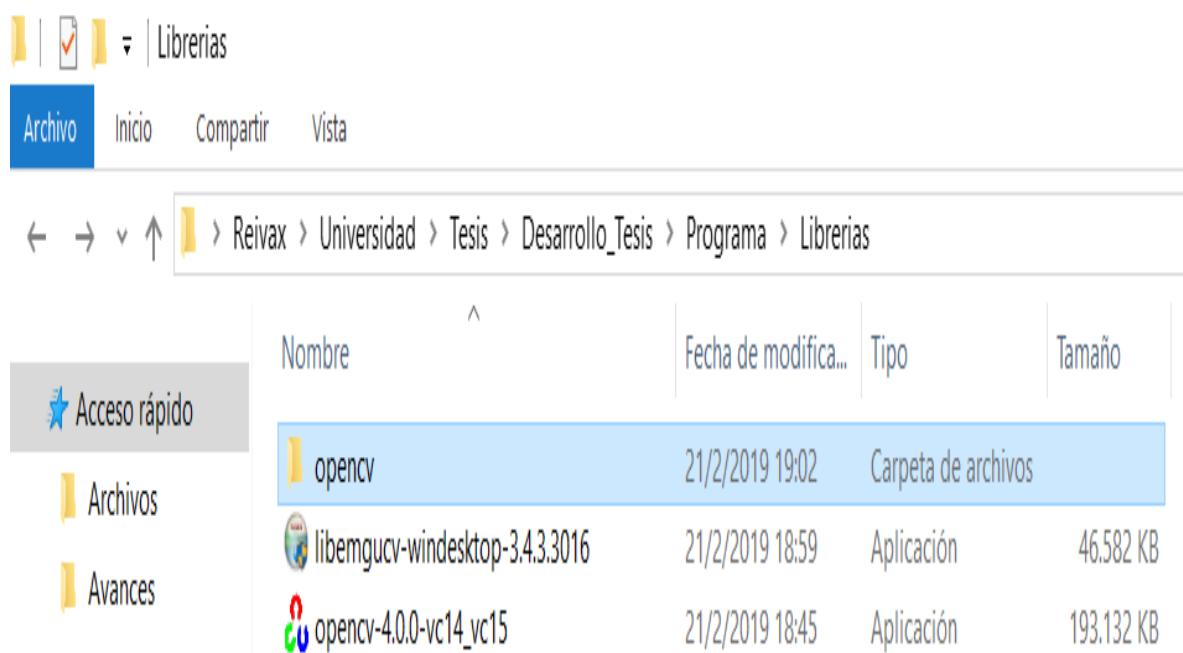


Figura 2.31. Carpeta creada con las librerías de OpenCV.

A continuación, se procede con la instalación de EmguCV; de igual manera que OpenCV, se ejecuta el instalador como administrador y se sigue los pasos que indica el instalador, luego se selecciona la ruta donde se quiere instalar, para este caso se recomienda dejar la ruta por defecto para evitar algún tipo de problema futuro, como se observa en la Figura 2.32.

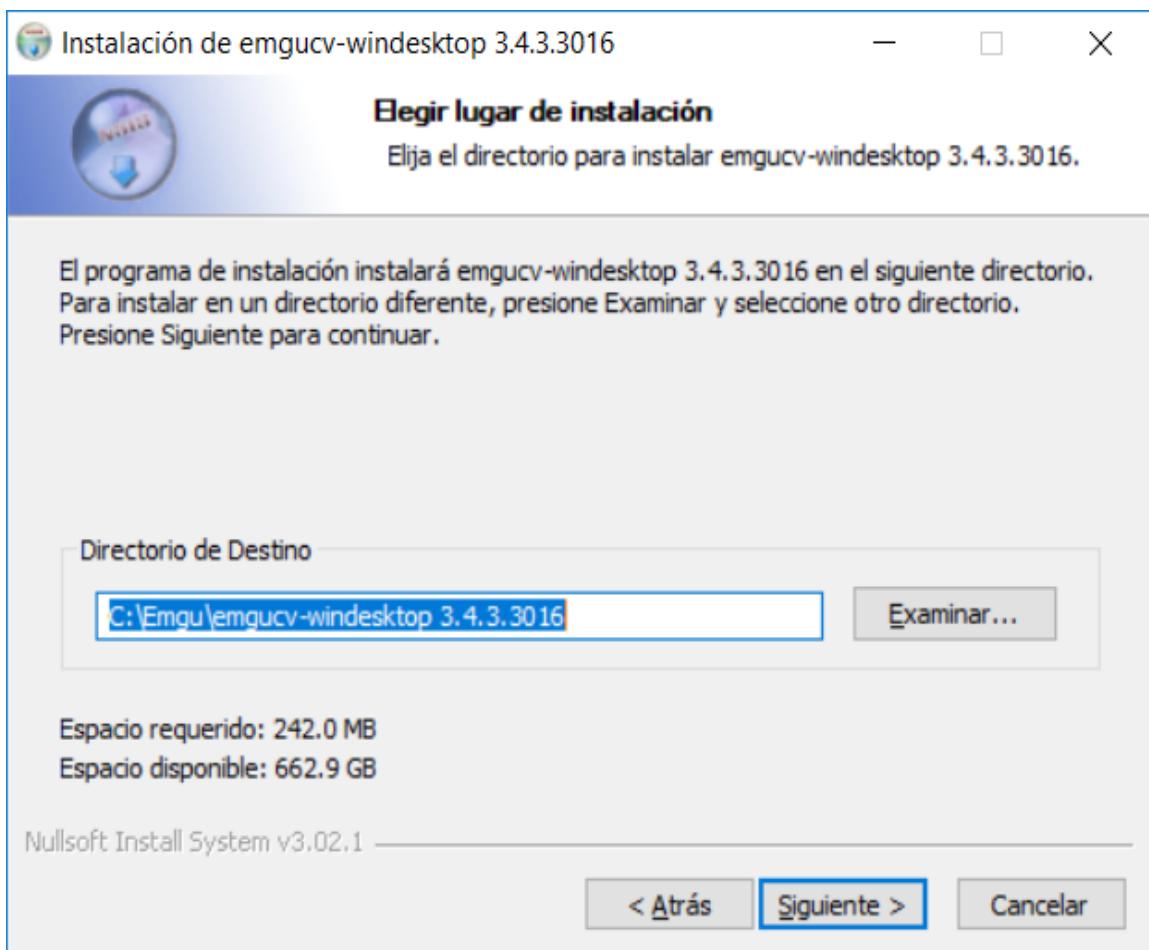


Figura 2.32. Ruta instalación archivo EmguCV.

En el proceso de instalación es posible que aparezca el mensaje de instalación del depurador EmguCV en Visual Studio. Se recomienda instalar este depurador en Visual Studio ya que ayudará a corregir errores en el programa que el propio Visual Studio no detecte, como se observa en la Figura 2.33.

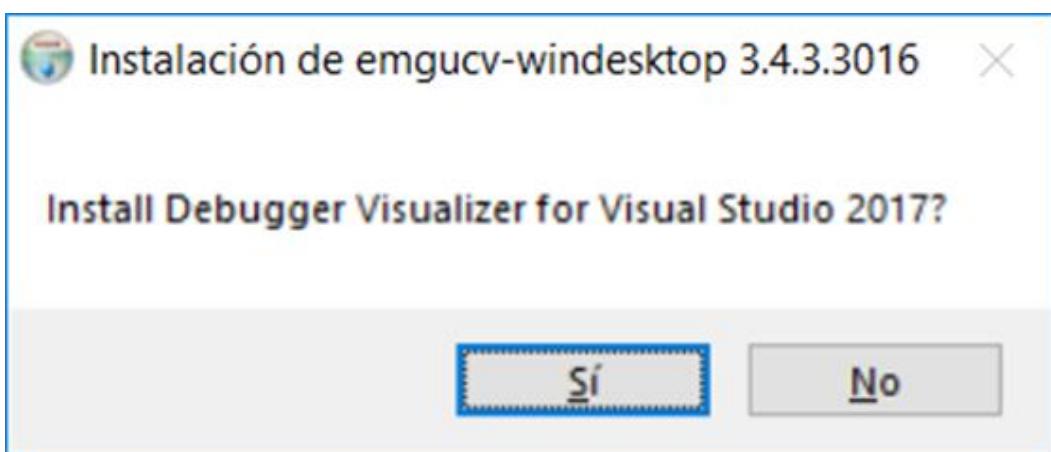


Figura 2.33. Depurador de EmguCV en Visual Studio.

Para poder ver las librerías instaladas, simplemente se busca la ruta en donde se instaló e igual que en OpenCV se puede encontrar varias librerías destinadas a diferentes proyectos y código útil para el desarrollo de los mismos.

Ahora, se procede con la instalación de Ozeki SDK, el instalador se puede descargar de su página oficial (http://www.ozeki.hu/index.php?owpn=1017&download_product_id=3). En esta página existen varias versiones para Windows y Linux, es recomendable bajarse la última versión como se observa en la Figura 2.34.

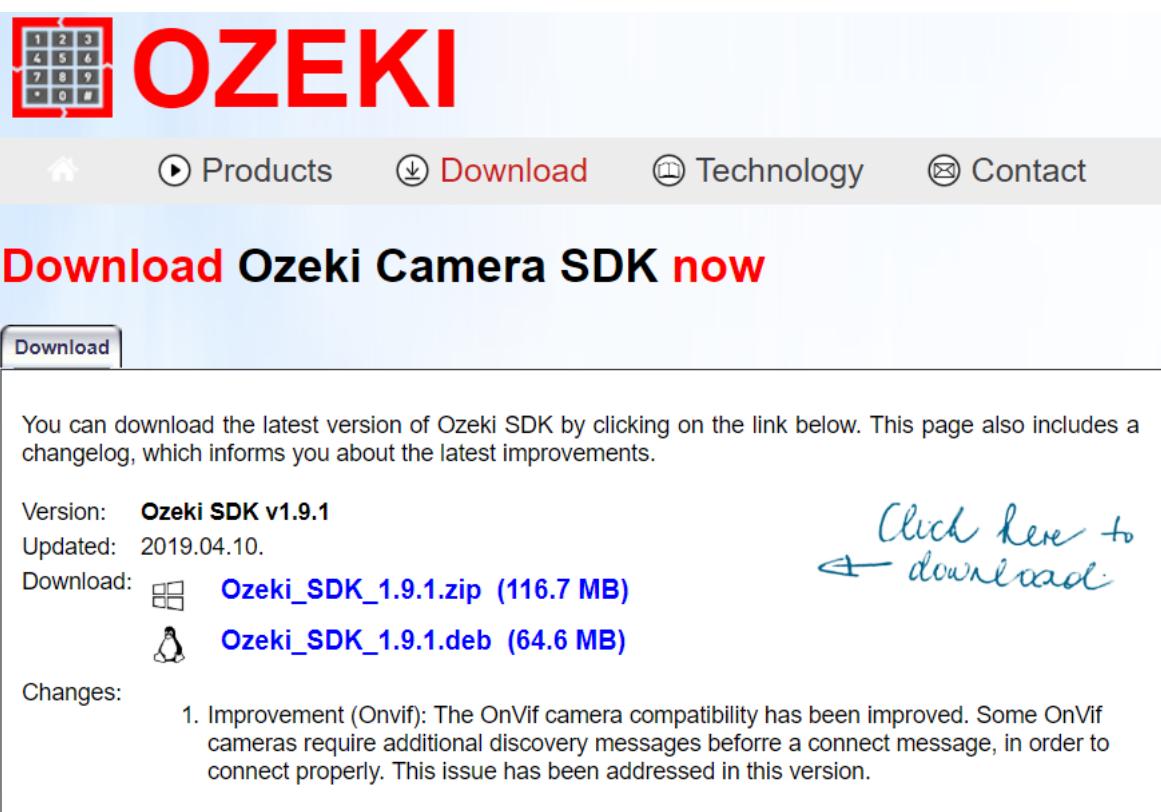


Figura 2.34. Descarga del SDK de Ozeki.

Una vez descargado el instalador se lo ejecuta como administrador y se sigue los pasos indicados por el instalador, se selecciona la ruta de instalación como se ve en la Figura 2.35 y finalmente se finaliza la instalación.

Al finalizar la instalación se puede observar en el escritorio del computador que se han creado varios accesos rápidos para los diferentes componentes de Ozeki tales como Voz sobre IP, Ozeki Camera, entre otros. El que interesa es el “Ozeki Camera SDK” ya que éste contiene las librerías, documentación y ejemplos para el desarrollo del proyecto como se puede observar en la Figura 2.36.

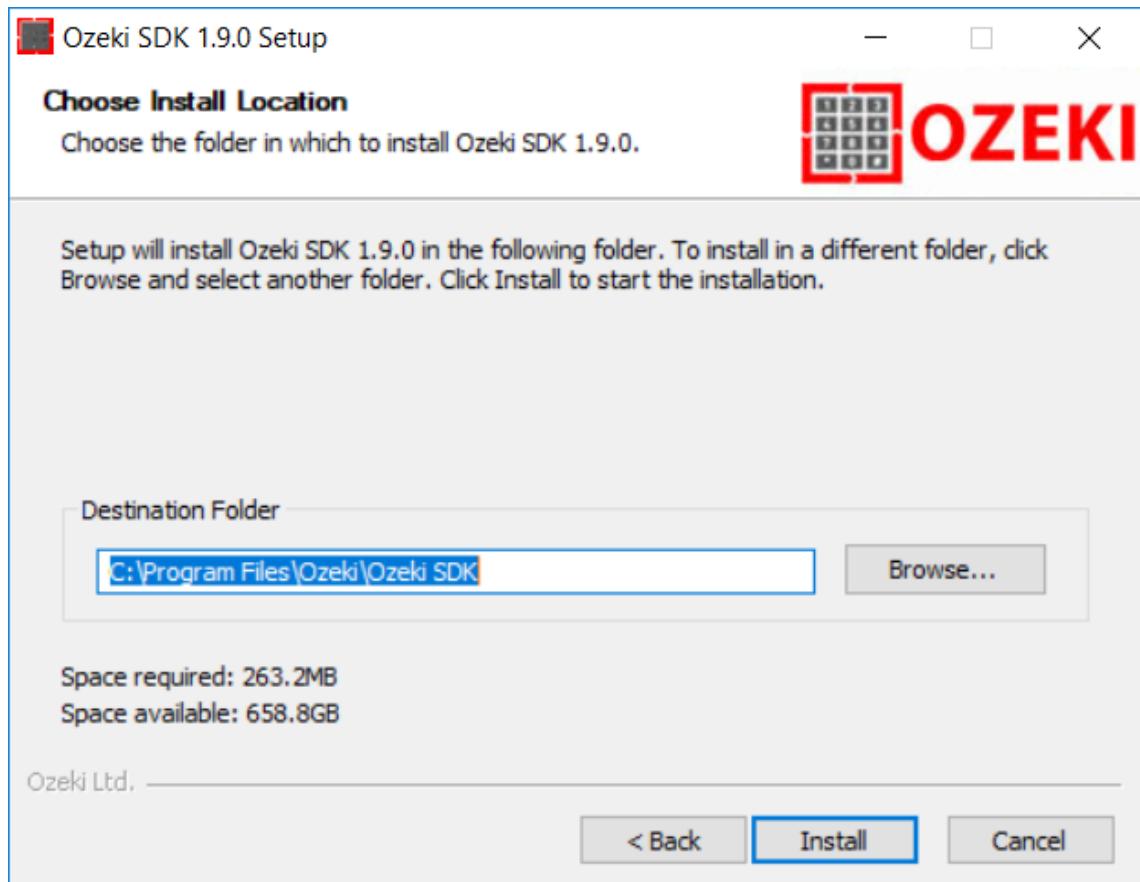


Figura 2.35. Ruta de instalación archivos Ozeki.

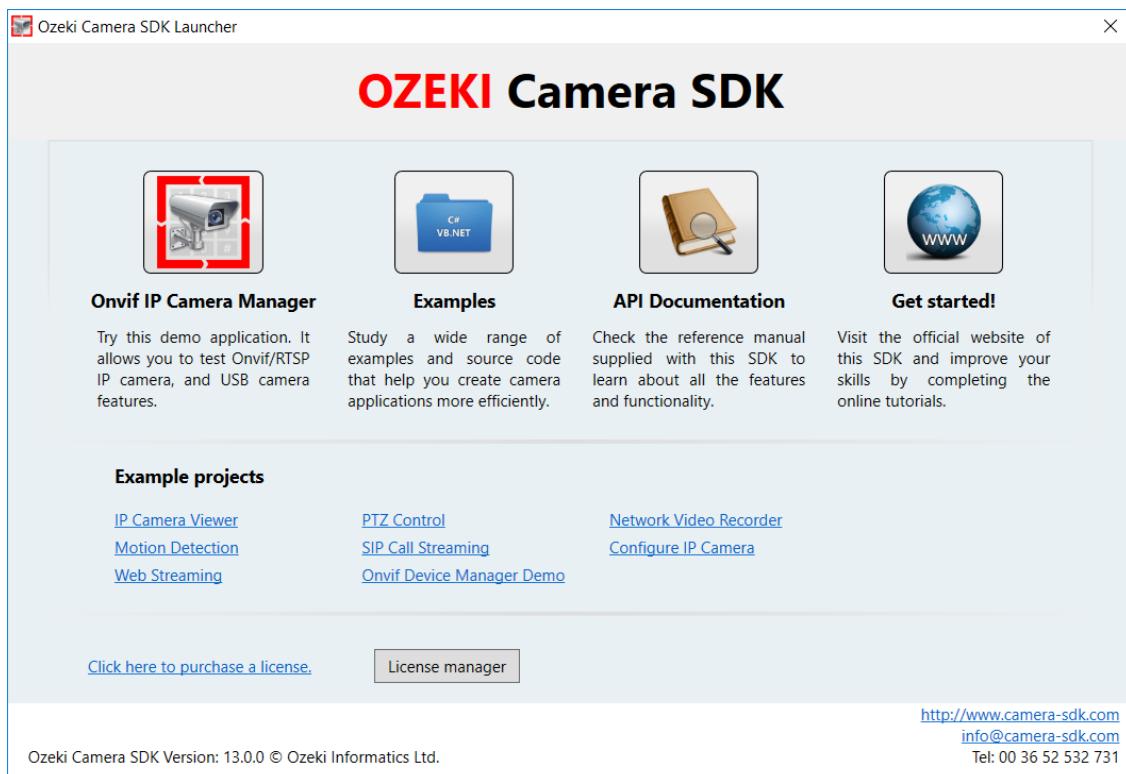


Figura 2.36. Interfaz del Ozeki Cámara SDK.

En este punto, corresponde la instalación de las librerías de ZKTeco; para ello se dirige a la página oficial de ZKTeco (<https://www.zktecolatinoamerica.com/descargas>). En esta página se descarga el SDK “ZKBioSecurity 3.1.5.0”, como muestra la Figura 2.37.

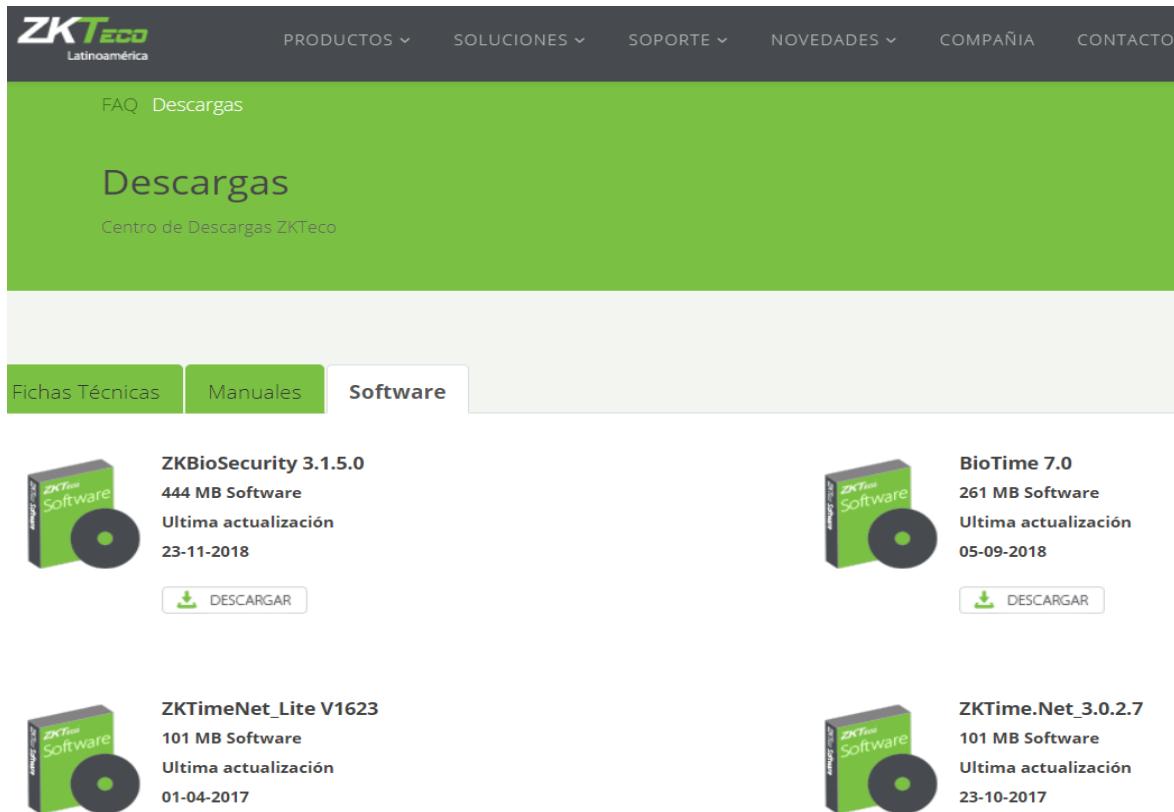


Figura 2.37. Descarga SDK ZKTeco.

Una vez descargado se ejecuta el instalador como administrador y se prosigue con los pasos como indica el instalador; se recomienda dejar las rutas de instalación por defecto. Este SDK es el cual permite a la aplicación de escritorio reconocer los dispositivos lectores de huellas dactilares, reconocer si están conectados o no. Se puede observar en la Figura 2.38 la instalación del SDK de ZKTeco.

Una vez finalizada la instalación, se tendrá que reiniciar el equipo para que se puedan efectuar los cambios realizados. Ver Figura 2.39.

Para poder unificar el SDK de ZKTeco con visual estudio es necesario descargar una librería llamada “Interop.ZKFPEngXControl.dll”, la cual permite reconocer las huellas dactilares dentro del entorno de Visual Studio; esta librería puede ser descargada de la siguiente dirección: <https://www.zhaodll.co/i/20130514/202289.html>, como muestra la Figura 2.40.

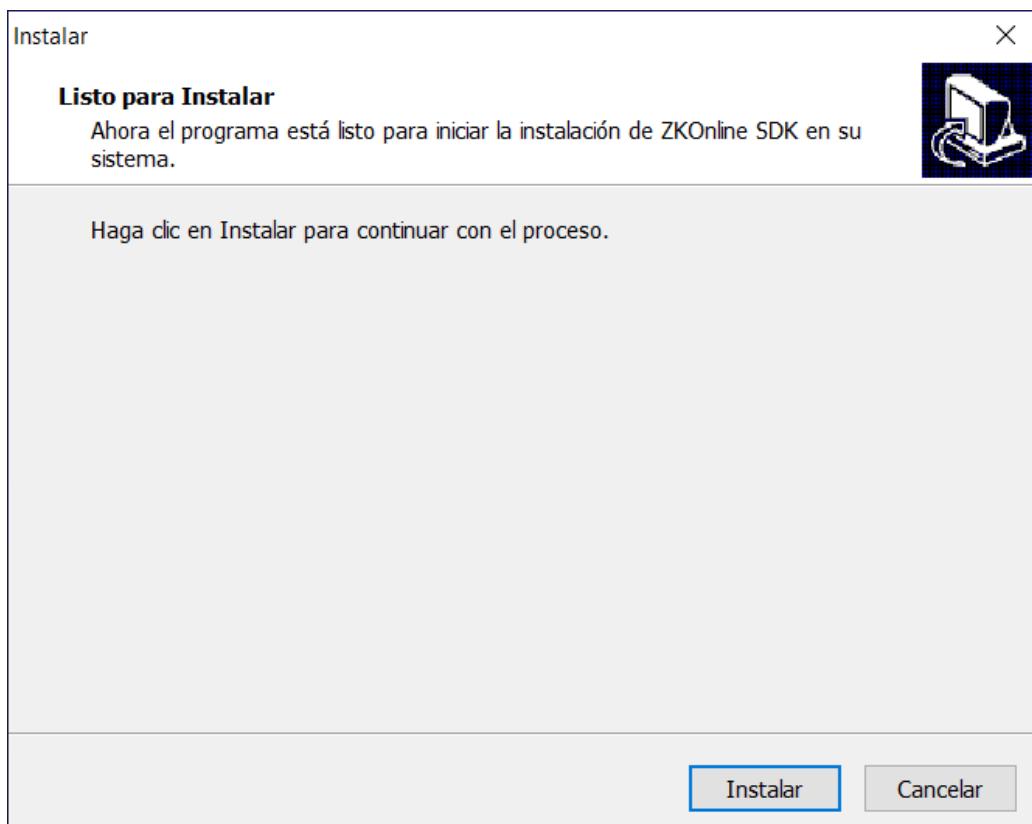


Figura 2.38. Instalación SDK ZKTeco.



Figura 2.39. Finalización instalación SDK ZKTeco.

Interop.ZKFPEngXControl.dll

[Download](#)

[To enter the download address list](#)

Using the zip file

Download Interop.ZKFPEngXControl.dll [To enter the download address list](#)

Extract Interop.ZKFPEngXControl.dll from .zip. We recommend that you extract

Interop.ZKFPEngXControl.dll to the installation directory of the program that is requesting
Interop.ZKFPEngXControl.dll.

Figura 2.40. Descarga librería Interop.ZKFPEngXControl.dll.

2.2.3. CREACIÓN DE LOS FORMULARIOS DE LA INTERFAZ GRÁFICA

El primer paso para la creación de los formularios de la interfaz gráfica es abrir Visual Studio y Crear un nuevo proyecto, como se muestra en la Figura 2.41. En el recuadro que se abre se busca una solución en blanco para Visual Studio, se da un nombre y la ruta donde se desea crear el proyecto y se presiona en aceptar, como se puede ver en la Figura 2.42.

Aquí se puede ver como se crea un proyecto en blanco, en la ruta y el nombre especificado anteriormente; este proyecto se encuentra vacío, es decir no existe ninguna clase o formulario en su interior. El siguiente paso es crear un nuevo proyecto; esto se hace seleccionando con clic derecho sobre la solución y se agrega un nuevo proyecto, como se observa en la Figura 2.43. En la nueva ventana que se abre se busca el lenguaje de Visual C# y la opción de “Aplicación de Windows Forms (.NET Framework)”, se da un nombre y a aceptar, lo mencionado se puede apreciar en la Figura 2.44.

Este nuevo elemento creado será aquel que albergue todos los elementos visuales de la aplicación de escritorio. Dentro de este se agregarán desde el cuadro de herramientas elementos tales como: Botones, CheckBox, ComboBox, DataGridView, GroupBox, Labels TextBox, etc., los mismos que permitirán la manipulación de la aplicación. La Figura 2.45, muestra la interfaz gráfica principal terminada. Cabe mencionar que, a cada elemento se le ha proporcionado un nombre y un texto; por ejemplo: elemento botón con el nombre: boton_Guardar y el texto: Guardar.

En las Figuras 2.46 a 2.50, respectivamente, se muestran las diferentes interfaces que componen la aplicación de escritorio estas interfaces corresponden a: log de eventos, administración de usuarios, agregar lectores, agregar teclados e inicio de sesión.

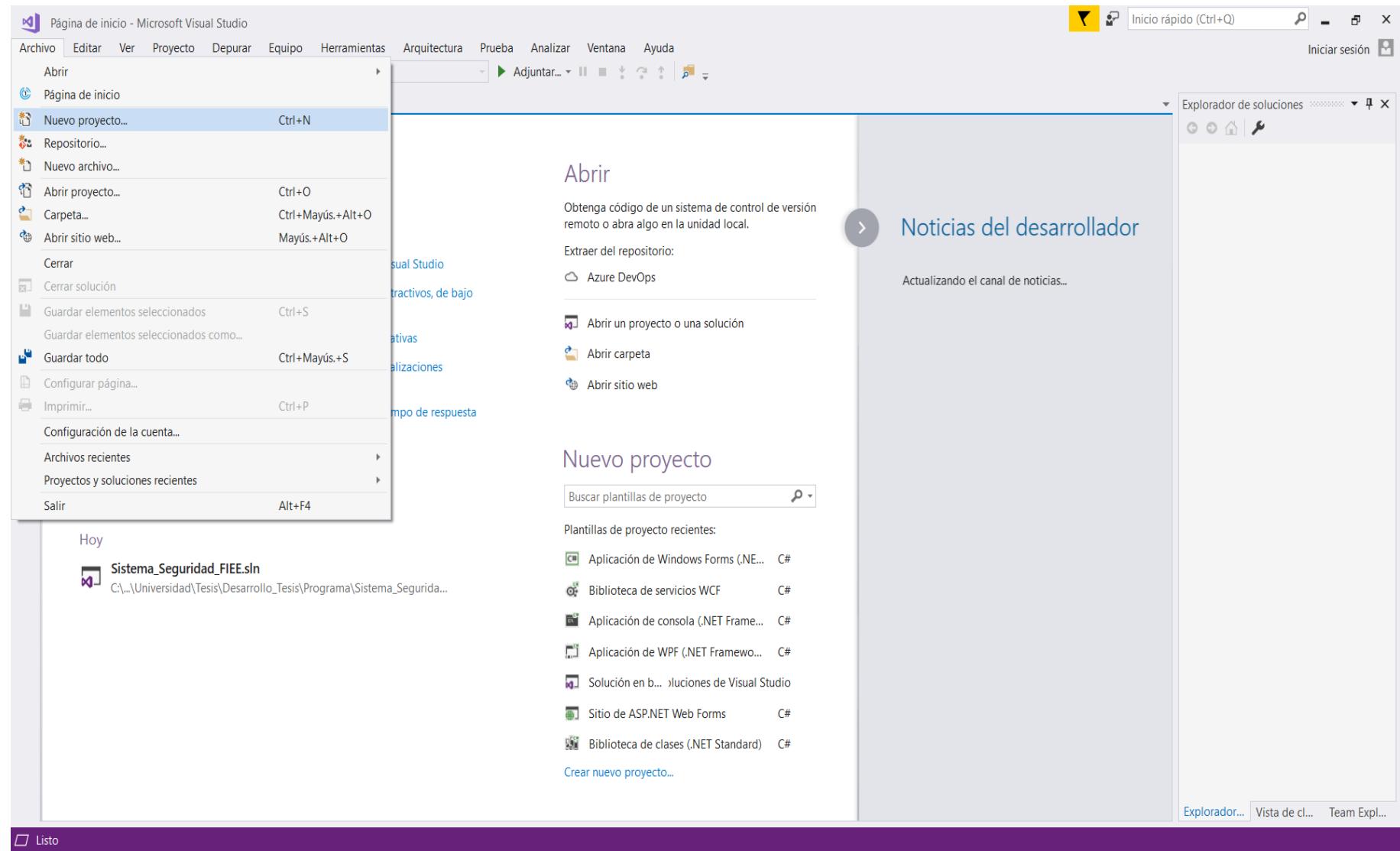


Figura 2.41. Creación de un nuevo proyecto en Visual Studio.

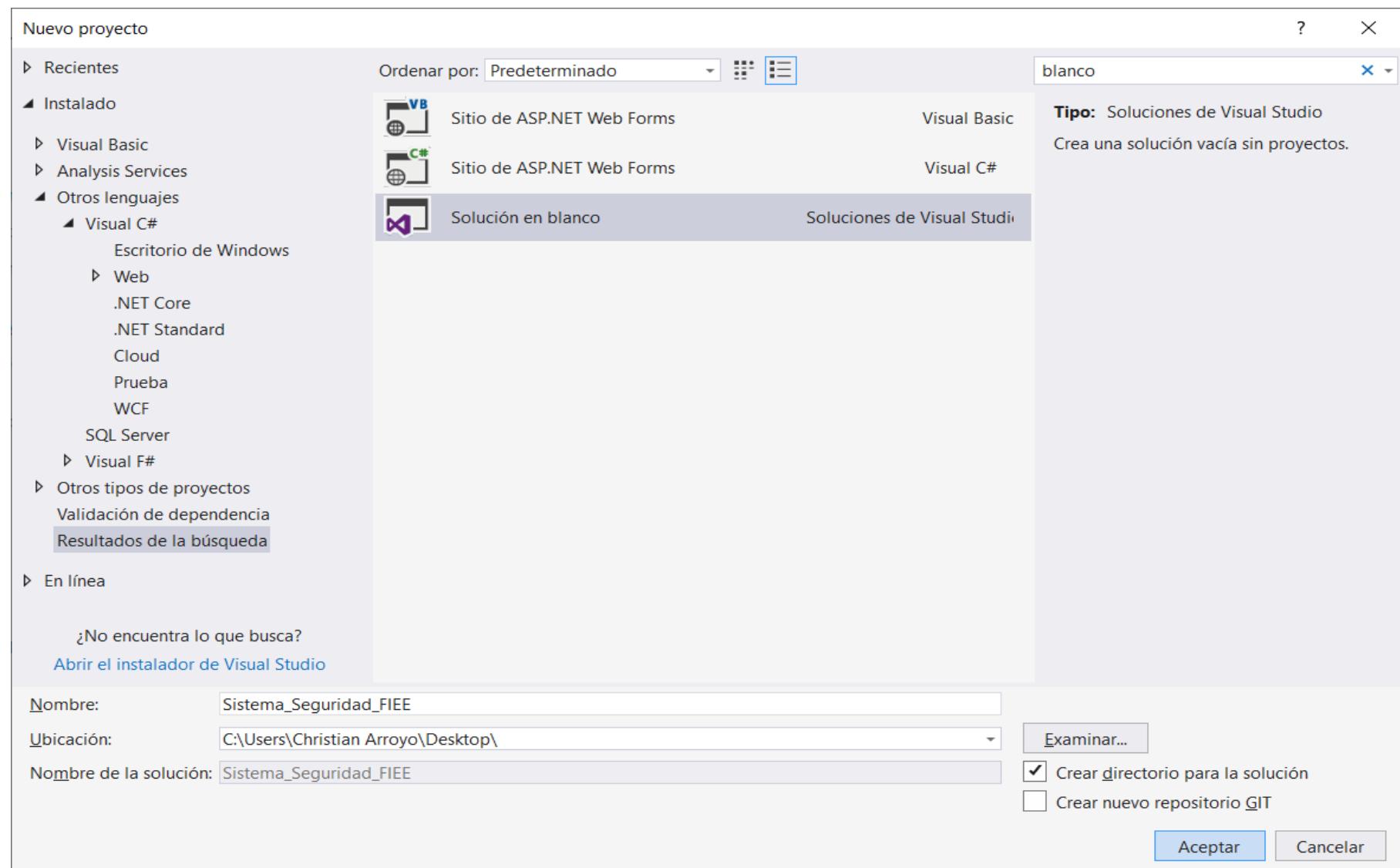


Figura 2.42. Creación de solución en blanco del nuevo proyecto.

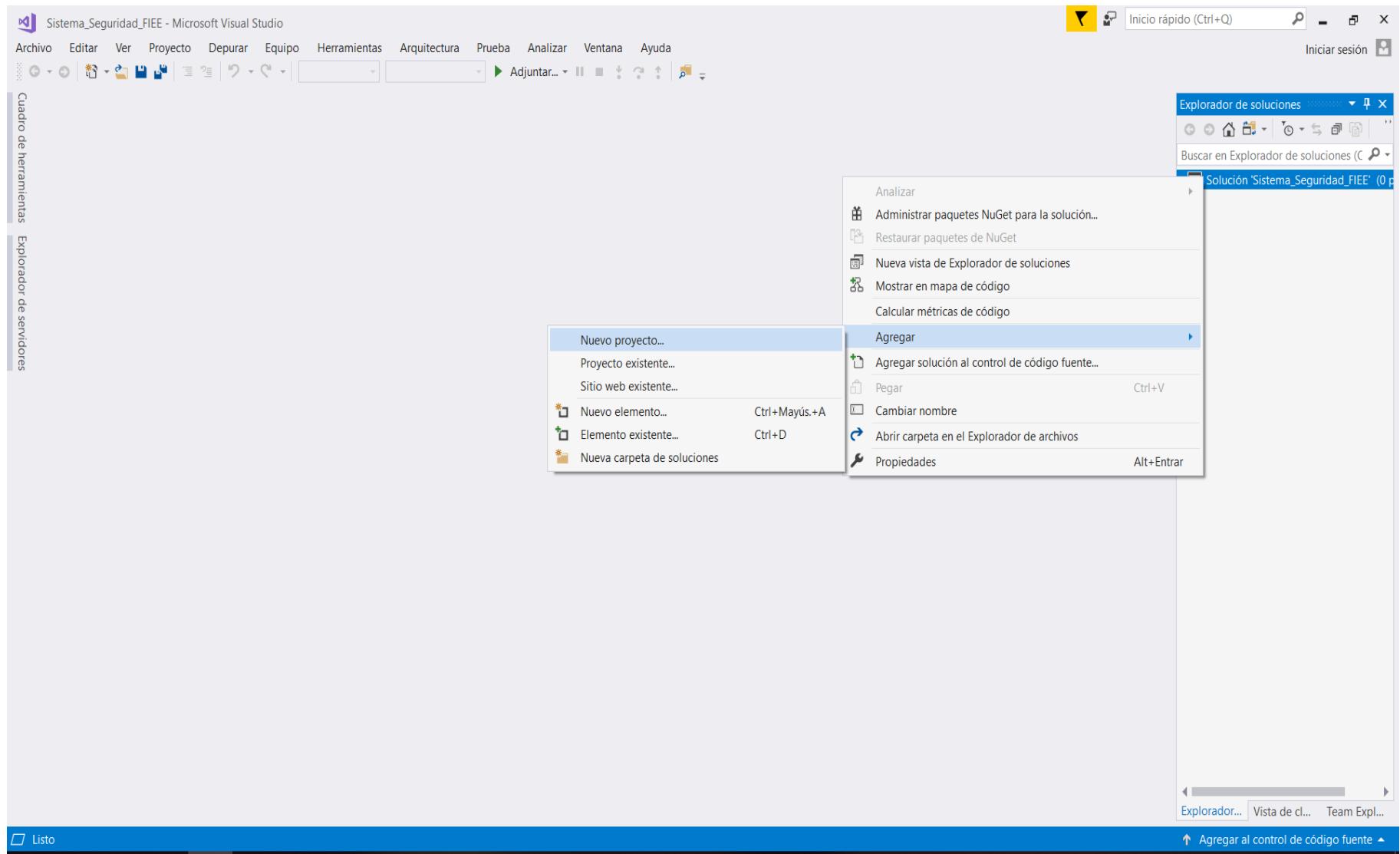


Figura 2.43. Creación de nuevo proyecto dentro de la solución.

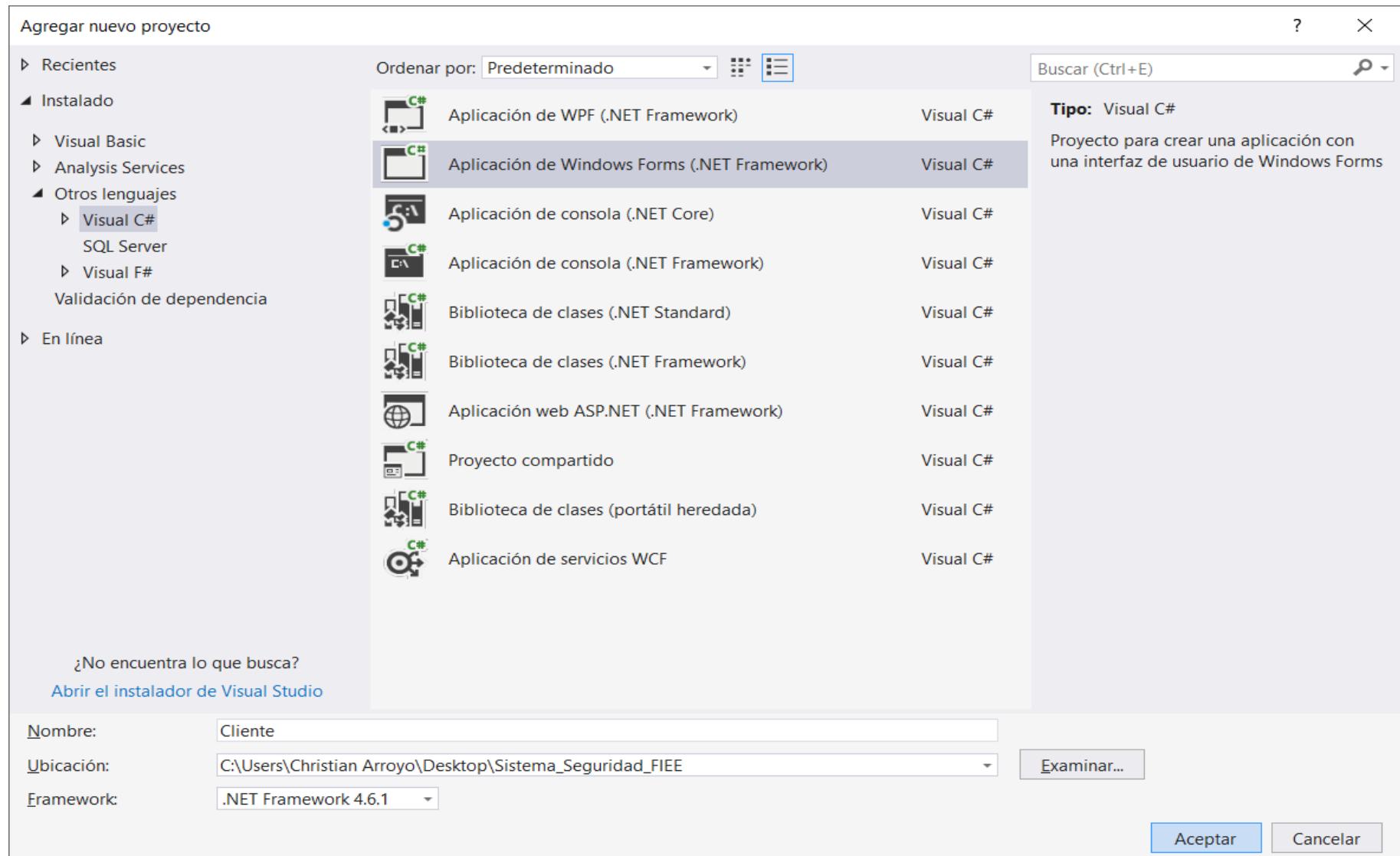


Figura 2.44. Creación nuevo formulario en el proyecto.

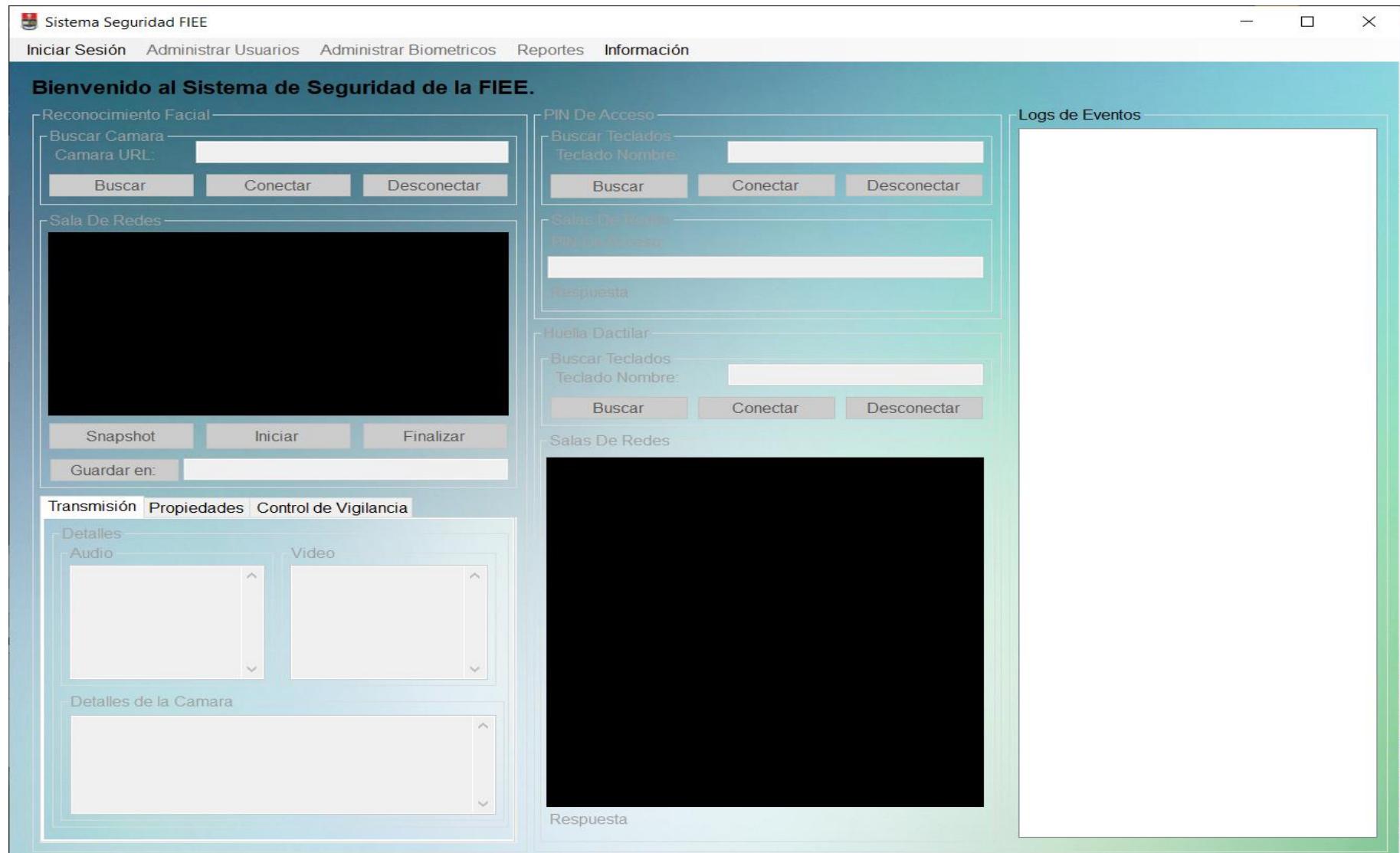


Figura 2.45. Interfaz gráfica del formulario principal.

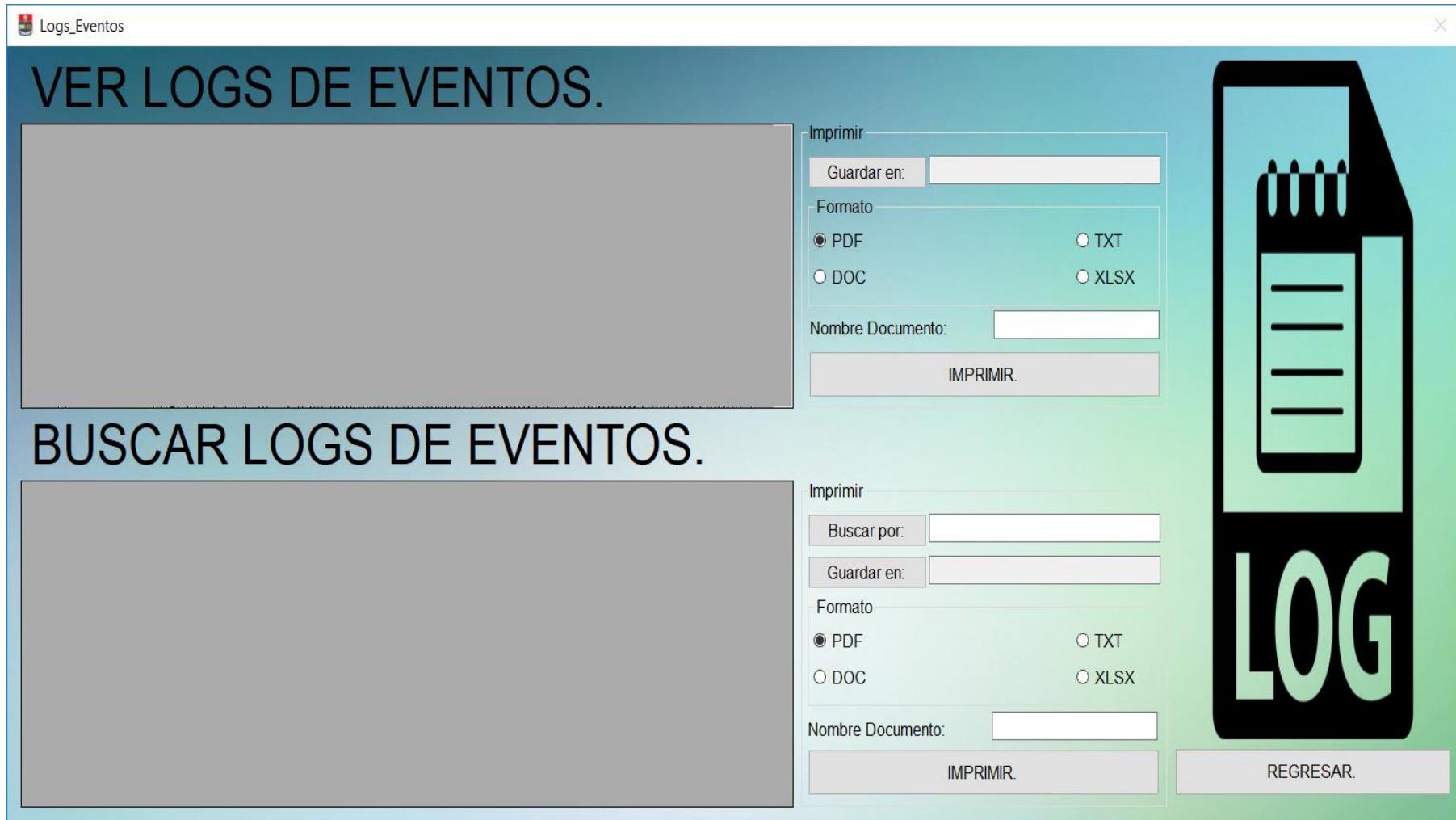


Figura 2.46. Interfaz gráfica del log de eventos.

Administracion_Usuarios X

CREAR NUEVA CUENTA.

ID: 6

CEDULA:

ALIAS:

NOMBRE:

APELLIDO:

CONTRASEÑA:

CONFIRMAR CONTRASEÑA:

TIPO:

DEPARTAMENTO:

CORREO:

PIN DE ACCESO:

HUELLA DACTILAR:

DETECCIÓN FACIAL:

TOMAR HUELLA

TOMAR FOTO BUSCAR FOTO

REGISTRAR ACTUALIZAR. ELIMINAR. REGRESAR.

Detección Facial: [Placeholder]

Buscar Foto: [Placeholder]

Rostros Detectados: 0 Guardar

Rostros Detectados: 0 Guardar

Habilitar

Huella Dactilar

Registrar Huella

Verificar Huella

Informacion: 0 Serial Dispositivo: 0 Registro Huella: Guardar Huella

Logs de Eventos

Figura 2.47. Interfaz gráfica de la administración de usuarios.

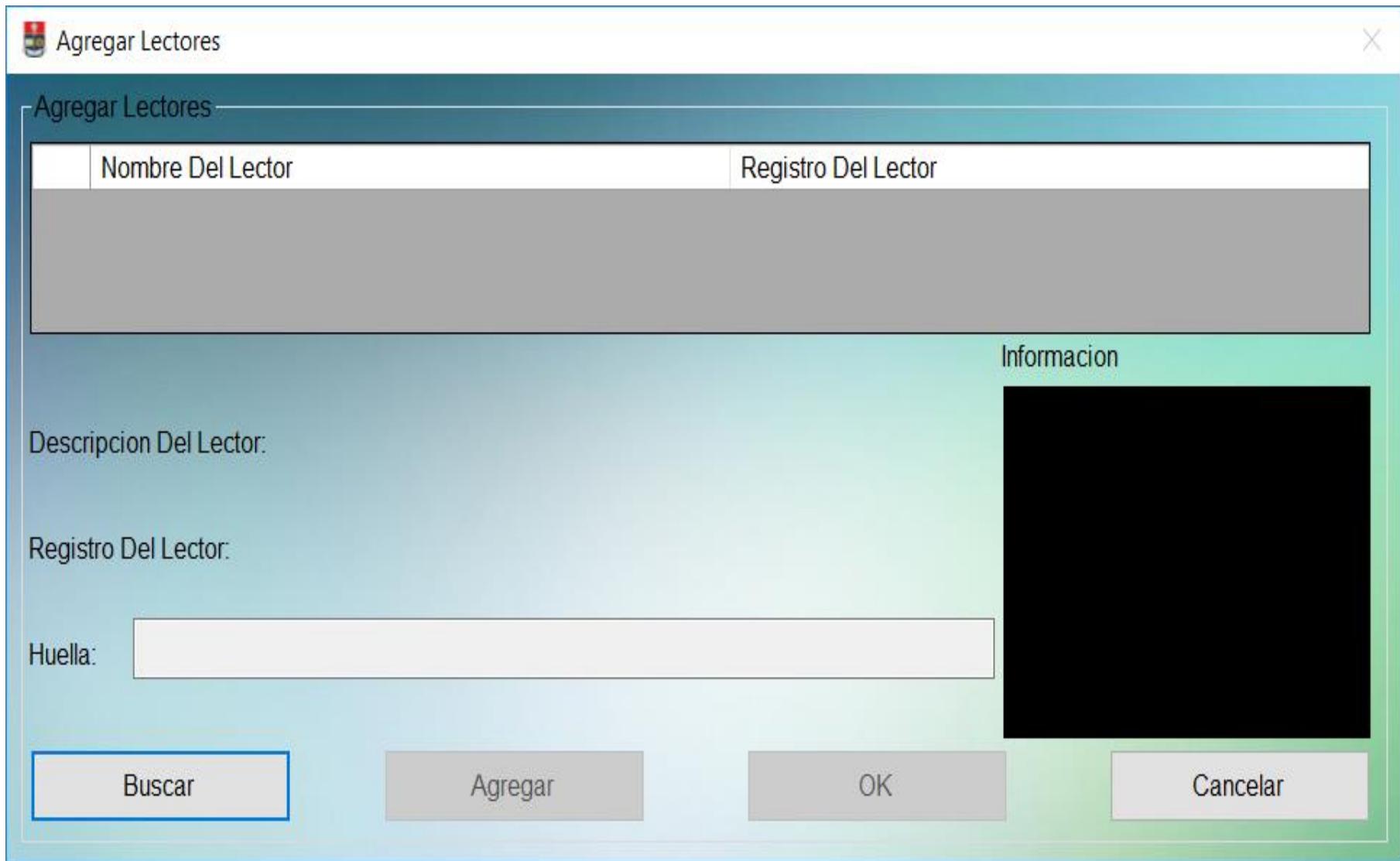


Figura 2.48. Interfaz gráfica para agregar lectores.

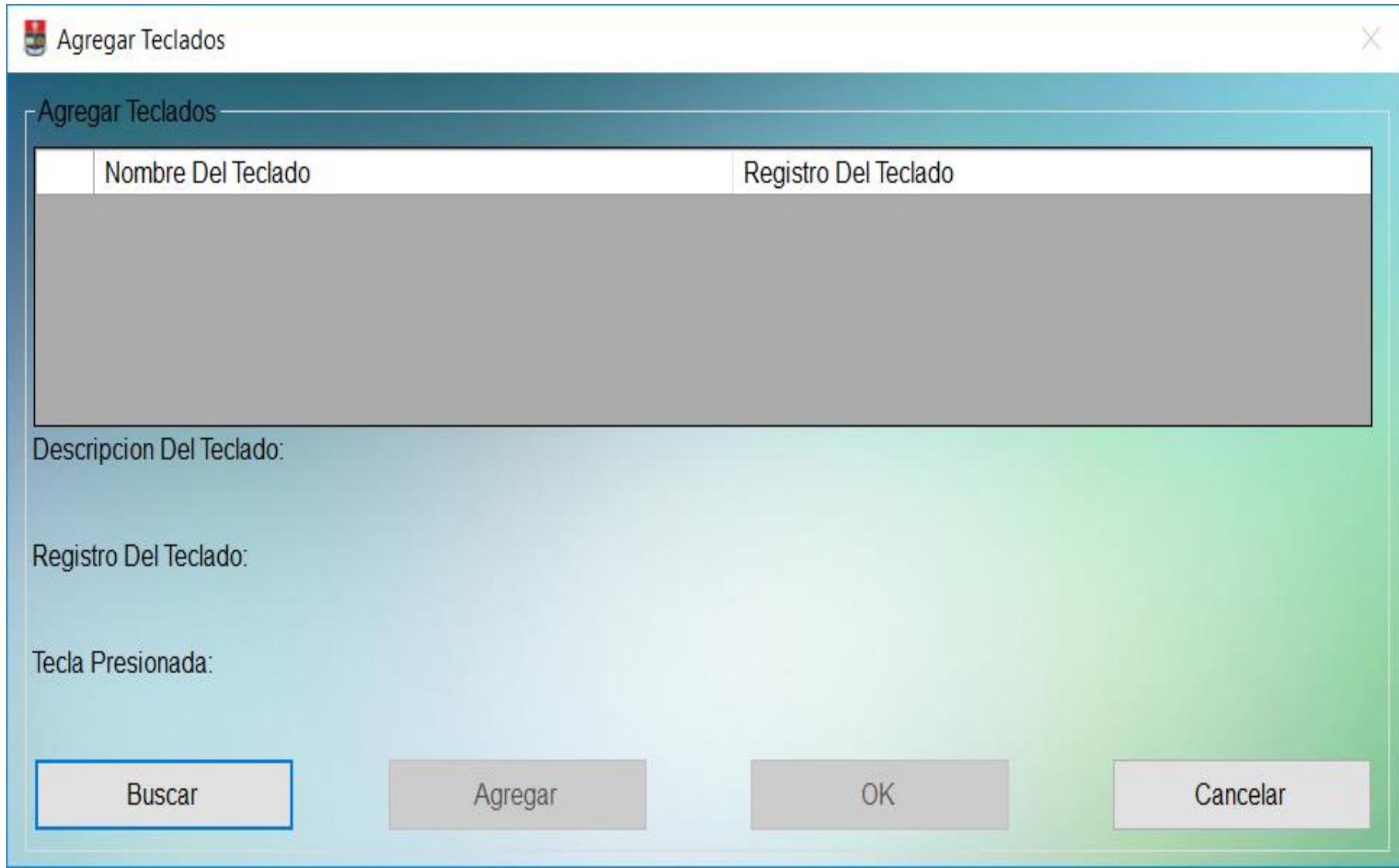


Figura 2.49. Interfaz gráfica para agregar teclados.



Figura 2.50. Interfaz gráfica del inicio de sesión.

2.2.4. IMPLEMENTACIÓN DE CADA MÓDULO

Los módulos involucrados en la aplicación de escritorio son: “Interfaz de Usuario”, “Lógica De Negocios”, “Acceso A Datos” y “Base De Datos”. Para el modulo Interfaz Gráfica no se requiere mayor programación por parte del usuario; ya que la plataforma es la que configura los elementos automáticamente al agregarlos, el usuario solo debe colocar los controles en una ubicación específica del formulario y agregarles un nombre para poder identificarlos.

Dentro de la Interfaz Gráfica la programación que se realizó es el controlar los elementos visuales; ya sea deshabilitándolos o habilitándolos según sea necesario y el controlar la salida del formulario mediante el ícono superior “X”. En el Código 2.1 se muestra parte del código para bloquear los elementos visuales y en el Código 2.2 se muestra la anulación del evento cerrar.

```

private void BloquearElementos()
{
    boton_DeshabilitarReconocimiento1.Enabled = false;
    boton_DeshabilitarReconocimiento2.Enabled = false;
    groupBox_ReconocimientoSala1.Enabled = false;
    groupBox_ReconocimientoSala2.Enabled = false;
    boton_DesconectarTeclado.Enabled = false;
    boton_DesconectarLector.Enabled = false;
    administrarBiometricos.Enabled = false;
    boton_ConectarTeclado.Enabled = false;
    boton_ConectarLector.Enabled = false;
    administrarUsuarios.Enabled = false;
    txt_NombreTeclado.Enabled = false;
    boton_Desconectar.Enabled = false;
    groupBox_Teclado1.Enabled = false;
    groupBox_Teclado2.Enabled = false;
    reportesEventos.Enabled = false;
    groupBox_Facial.Enabled = false;
    groupBox_Huella.Enabled = false;
    boton_Conectar.Enabled = false;
    groupBox_Sala1.Enabled = false;
    groupBox_Sala2.Enabled = false;
    txt_CamaraURL.Enabled = false;
    logoutSistema.Enabled = false;
    groupBox_Pin.Enabled = false;
}

```

Código 2.1. Método para bloquear los controles visuales.

```

private const int noBotonCerrar = 0x200;
protected override CreateParams CreateParams
{
    get
    {
        CreateParams cerrar = base.CreateParams;
        cerrar.ClassStyle = cerrar.ClassStyle | noBotonCerrar;
        return cerrar;
    }
}

```

Código 2.2. Anulación del evento cerrar.

Para la “Lógica De Negocios” se procedió, primero, a establecer la comunicación con el servidor; así el cliente puede realizar consultas a la base de datos. En el Código 2.3 se muestra el código para iniciar y finalizar conexión con el servidor.

El Código 2.4, permite reconocer a los teclados que se encuentran conectados al computador; adicionalmente, permite recibir los datos enviados por el teclado a la aplicación de escritorio.

```

private ChannelFactory<IObjetoRemoto> canal = new ChannelFactory<IObjetoRemoto>
private Thread servidor = new Thread(Servidor.Main);
private IObjetoRemoto interfaz;

private void FinalizarServidor()
{
    (interfaz as ICommunicationObject).Close();
    Servidor.StopServer();
}

private void IniciarServidor()
{
    interfaz = canal.CreateChannel();
    servidor.IsBackground = true;
    Thread.Sleep(1000);
    servidor.Start();
}

```

Código 2.3. Conexión con el servidor.

```

private void Boton_Buscar_Click(object sender, EventArgs e)
{
    DialogResult busqueda = MessageBox.Show("Antes de buscar u
    if (busqueda == DialogResult.OK)
    {
        TecladosGuardados();
        boton_Buscar.Enabled = false;
        boton_Agregar.Enabled = true;
        dispositivo = new Eventos_Teclado(Handle);
        numeroTeclados = dispositivo.NumeroDispositivos();
        dispositivo.TeclaPresionada += new Eventos_Teclado.Con
    }
    else
    {
        return;
    }
}

protected override CreateParams CreateParams
{
    get
    {
        CreateParams cerrar = base.CreateParams;
        cerrar.ClassStyle = cerrar.ClassStyle | noBotonCerrar;
        return cerrar;
    }
}

```

Código 2.4. Reconocimiento de teclados y procesamiento de mensajes.

En el módulo de “Lógica De Negocios” se estableció, también, la programación necesaria para detectar y reconocer los rostros y huellas dactilares de los usuarios. En el Código 2.5 se muestra el código para la detección y reconocimiento facial y en el Código 2.6 se presenta el código para la detección y reconocimiento dactilar.

```

private void Capturadora(object sender, EventArgs e)
{
    videoRostro = imagen.QueryFrame().Resize(320, 240, INTER.CV_INTER_CUBIC);
    escalaGrises = videoRostro.Convert<Gray, byte>();
    MCvAvgComp[][] rostroDetectado = escalaGrises.DetectHaarCascade(deteccionRostro, 1.2, 10, HAAR_DETECTION_TYPE.DO_CANNY_PRUNING, new Size(20, 20));
    foreach (MCvAvgComp rostro in rostroDetectado[0])
    {
        numeroNombres = numeroNombres + 1;
        imagenGris = videoRostro.Copy(rostro.rect).Convert<Gray, byte>().Resize(100, 100, INTER.CV_INTER_CUBIC);
        videoRostro.Draw(rostro.rect, new Bgr(Color.Red), 2);
        if (imagenesEntrenamiento.ToArray().Length != 0)
        {
            MCvTermCriteria criterio = new MCvTermCriteria(contadorEntrenamiento, 0.001);
            EigenObjectRecognizer reconocedor = new EigenObjectRecognizer(imagenesEntrenamiento.ToArray(), nombres Usuarios.ToArray(), 3000, ref criterio);
            nombre = reconocedor.Recognize(imagenGris);
            videoRostro.Draw(nombre, ref font, new Point(rostro.rect.X - 2, rostro.rect.Y - 2), new Bgr(Color.LightGreen));
        }
        nombrePersonas.Add("");
        nombrePersonas[numeroNombres - 1] = nombre;
        nombrePersonas.Add("");
        etiquetaCantidadRostrosDetectados.Text = rostroDetectado[0].Length.ToString();
        escalaGrises.ROI = rostro.rect;
        MCvAvgComp[][] eyesDetected = escalaGrises.DetectHaarCascade(deteccionOjos, 1.1, 10, HAAR_DETECTION_TYPE.DO_CANNY_PRUNING, new Size(20, 20));
        escalaGrises.ROI = Rectangle.Empty;
        foreach (MCvAvgComp ey in eyesDetected[0])
        {
            Rectangle eyeRect = ey.rect;
            eyeRect.Offset(rostro.rect.X, rostro.rect.Y);
            videoRostro.Draw(eyeRect, new Bgr(Color.Blue), 2);
        }
    }
    numeroNombres = 0;
    for (int nnn = 0; nnn < rostroDetectado[0].Length; nnn++)
    {
        nombres = nombres + nombrePersonas[nnn] + ", ";
    }
    reconocimientoRostro.Image = videoRostro;
    nombres = "";
    nombrePersonas.Clear();
}

```

Código 2.5. Detección y reconocimiento facial.

```

dispositivoDactilar.OnFeatureInfo += Informacion_Huella;
dispositivoDactilar.OnImageReceived += Huella_Recibida;
dispositivoDactilar.OnEnroll += Huella_Enrolada;
if (dispositivoDactilar.InitEngine() == 0)
{
    dispositivoDactilar.FPEngineVersion = "9";
    dispositivoDactilar.EnrollCount = 3;
    txt_SerialDispositivo.Text += " " + dispositivoDactilar.SensorSN;
    MostrarInformacion("Dispositivo Conectado Correctamente.");
}
else
{
    MessageBox.Show("No hay un lector dactilar conectado. Conecte uno");
    dispositivoDactilar.OnFeatureInfo -= Informacion_Huella;
    dispositivoDactilar.OnImageReceived -= Huella_Recibida;
    dispositivoDactilar.OnEnroll -= Huella_Enrolada;
    boton_VerificarHuella.Enabled = false;
    boton_RegistrarHuella.Enabled = true;
    boton_DetectarHuella.Enabled = true;
    groupBox_Dactilar.Enabled = false;
    pictureBox_Huella.Image = null;
}

```

Código 2.6. Detección y reconocimiento dactilar.

Finalmente, la “Lógica De Negocios” permite comunicar con la base de datos para realizar una consulta SQL, en el Código 2.7 se puede observar, como ejemplo, una petición realizada a la base de datos para registrar a un usuario.

```

objetoremotoUsuario.departamentoUsuario = comboBox_Departamento.SelectedItem.ToString();
objetoremotoUsuario.identificadorUsuario = Convert.ToInt32(txt_Id.Text);
objetoremotoUsuario.tipoUsuario = comboBox_Tipo.SelectedItem.ToString();
objetoremotoUsuario.cedulaUsuario = Convert.ToInt32(txt_Cedula.Text);
objetoremotoUsuario.pinUsuario = Convert.ToInt32(txt_Pin.Text);
objetoremotoUsuario.passwordUsuario = txt_Contraseña.Text;
objetoremotoUsuario.apellidoUsuario = txt_Apellido.Text;
objetoremotoUsuario.nombreUsuario = txt_Nombre.Text;
objetoremotoUsuario.correoUsuario = txt_Correo.Text;
objetoremotoUsuario.huellaUsuario = txt_Huella.Text;
objetoremotoUsuario.aliasUsuario = txt_Alias.Text;
objetoremotoUsuario.rostroUsuario = rostro;
(objetoremotoUsuario as IObjetoRemoto).AgregarUsuario(recibirDatosUsuario);

```

Código 2.7. Registrar un usuario en la base de datos.

Dentro del “Acceso a Datos” se encuentra el servidor, el cual permite la comunicación entre el cliente y la base de datos, el código del servidor se muestra en el Código 2.8.

```
public class Servidor
{
    private static System.Threading.AutoResetEvent eventoTerminacion = new System.Threading.AutoResetEvent(false);

    public static void Main()
    {
        ServiceHost anfitrion = new ServiceHost(typeof(ObjetoRemoto));
        anfitrion.AddServiceEndpoint(typeof(IObjetoRemoto), new NetTcpBinding(), "net.tcp://localhost:8080");
        anfitrion.Open();
        eventoTerminacion.WaitOne();
        Console.WriteLine();
        anfitrion.Close();
        Console.WriteLine();
    }

    public static void StopServer()
    {
        eventoTerminacion.Set();
    }
}
```

Código 2.8. Iniciar el servidor.

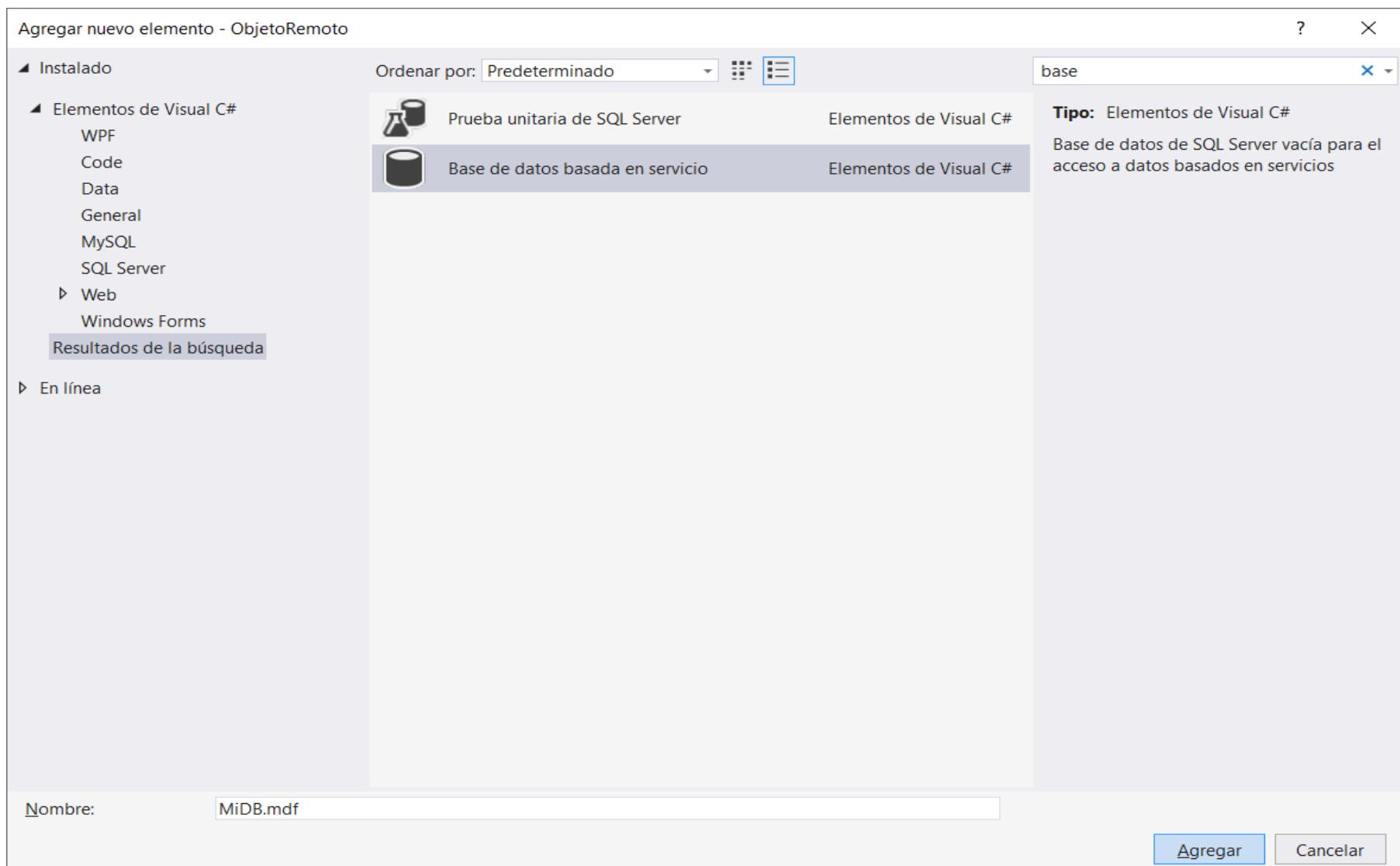


Figura 2.51. Creación de la base de datos.

Como se mencionó anteriormente dentro del “Acceso a Datos” se encuentra la base de datos; para agregar una base de datos al proyecto, simplemente se da a “Agregar Nuevo Elemento” y se busca “Base de datos basada en servicio”, se coloca un nombre y una ruta y se acepta. Se puede observar como agregar una base de datos en la Figura 2.51.

Una vez creada la base de datos corresponde a establecer la conexión, para ello dentro del entorno de Visual Studio se busca la opción “Explorador de servidores”; una vez ubicado, se agrega una nueva conexión buscando la base de datos en el lugar donde fue creada. Se realiza una prueba de conexión para verificar el estado de la base de datos, como se aprecia en la Figura 2.52.

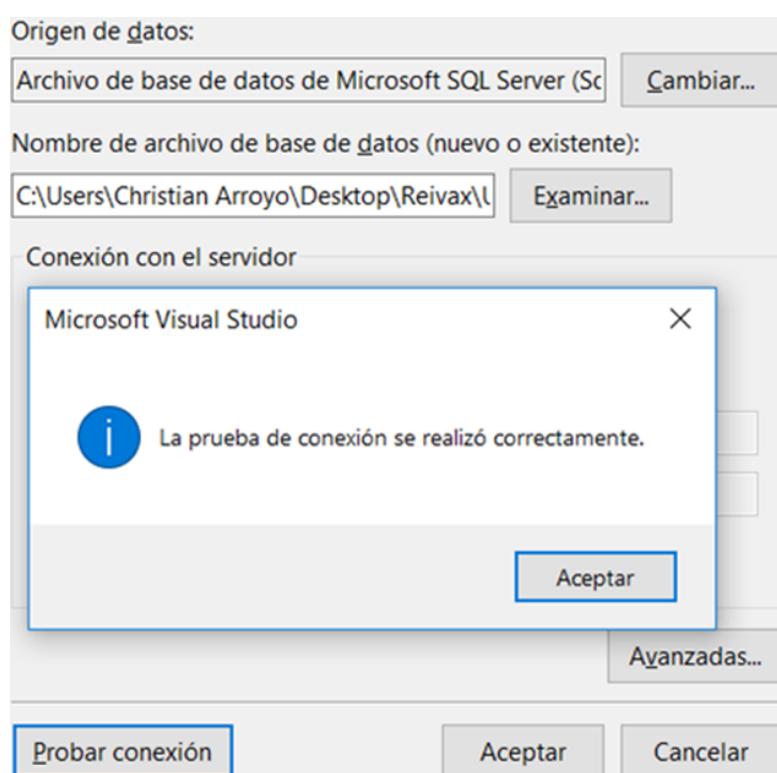


Figura 2.52. Prueba de conexión a la base de datos.

Una vez enlazada la base de datos a Visual Studio se procede a crear las tablas que compondrán la base de datos. En el Código 2.9 se muestra el código para crear la tabla Usuarios.

Con la creación de las tablas finalizadas, el siguiente paso es enlazar dichas tablas al código de la aplicación; para ello se necesita crear unas clases en el proyecto “ObjetoRemoto” con el mismo nombre de las tablas creadas en la base de datos. En el Código 2.10 se muestra el código empleado en la tabla Usuarios.

```

CREATE TABLE [dbo].[Usuarios] (
    [Id]          INT            NOT NULL,
    [Cedula]      INT            NOT NULL,
    [Alias]        VARCHAR (50)   NOT NULL,
    [Nombre]       VARCHAR (50)   NOT NULL,
    [Apellido]     VARCHAR (50)   NOT NULL,
    [Password]    VARCHAR (50)   NOT NULL,
    [Tipo]         VARCHAR (50)   NOT NULL,
    [Departamento] VARCHAR (50)   NOT NULL,
    [Correo]       VARCHAR (50)   NOT NULL,
    [Pin]          INT            NOT NULL,
    [Huella]       VARCHAR (MAX)  NOT NULL,
    [Rostro]       VARBINARY (MAX) NOT NULL,
    PRIMARY KEY CLUSTERED ([Cedula] ASC)
);

```

Código 2.9. Creación de la tabla Usuarios.

```

[Serializable]
[Table(Name = "Usuarios")]
public class Usuarios
{
    [Column]
    public int Id;
    [Column(IsPrimaryKey = true)]
    public int Cedula;
    [Column]
    public string Alias;
    [Column]
    public string Nombre;
    [Column]
    public string Apellido;
    [Column]
    public string Password;
    [Column]
    public string Tipo;
    [Column]
    public string Departamento;
    [Column]
    public string Correo;
    [Column]
    public int Pin;
    [Column]
    public string Huella;
    [Column]
    public Binary Rostro;

    public Usuarios(string ini_nombre, string ini_apellido, string ini_huella)...
    public Usuarios(int ini_id, string ini_alias, string ini_password)...
    public Usuarios(string ini_nombre, string ini_apellido)...
    public Usuarios(string ini_correo)...
}

```

Código 2.10. Creación de la tabla Usuarios en Visual Studio.

En este punto se procede a crear la cadena de conexión con la base de datos dentro del ObjetoRemoto. Esto permite realizar los cambios a la base de datos; esto se muestra en el Código 2.11.

```
public Table<Usuarios> Usuarios;
public Table<Logs> Logs;
public MiDB() : base(@"Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=
```

Código 2.11. Cadena de conexión a la base de datos.

Finalmente, si se desea realizar una consulta a la base de datos es necesario crear las interfaces (Servicios) en IObjetoRemoto, como se ve en el Código 2.12 y las consultas en ObjetoRemoto como se observa en el Código 2.13. En el Código 2.14 se muestra, como ejemplo, una consulta hecha por el administrador para registrar a un nuevo usuario.

```
[ServiceContract]
public interface IObjetoRemoto
{
    [OperationContract]
    List<Usuarios> ObtenerAdministradores(List<Usuarios> lista_usuario);
    [OperationContract]
    List<Usuarios> ConsultaDatosUsuario(List<Usuarios> lista_usuario);
    [OperationContract]
    List<Usuarios> ConsultaPinUsuario(List<Usuarios> lista_usuario);
    [OperationContract]
    List<Usuarios> ModificarUsuario(List<Usuarios> lista_usuario);
    [OperationContract]
    List<Usuarios> MostrarUsuarios(List<Usuarios> lista_usuario);
    [OperationContract]
    List<Usuarios> CompararUsuario(List<Usuarios> lista_usuario);
    [OperationContract]
    List<Usuarios> EliminarUsuario(List<Usuarios> lista_usuario);
    [OperationContract]
    List<Usuarios> CompararCedula(List<Usuarios> lista_usuario);
    [OperationContract]
    List<Usuarios> CompararCorreo(List<Usuarios> lista_usuario);
    [OperationContract]
    List<Usuarios> AgregarUsuario(List<Usuarios> lista_usuario);
    [OperationContract]
    List<Usuarios> ObtenerHuellas(List<Usuarios> lista_usuario);
    [OperationContract]
    List<Usuarios> CompararPin(List<Usuarios> lista_usuario);
    [OperationContract]
    List<Logs> MostrarEventos(List<Logs> lista_eventos);
    [OperationContract]
    List<Logs> AgregarEvento(List<Logs> lista_eventos);
    [OperationContract]
    int ObtenerID_Usuario(int id);
    [OperationContract]
    int ObtenerID_Eventos(int id);
}
```

Código 2.12. Agregación de los servicios.

```
public List<Usuarios> ObtenerAdministradores(List<Usuarios> lista_usuario)...
```

```
public List<Usuarios> ConsultaDatosUsuario(List<Usuarios> lista_usuario)...
```

```
public List<Usuarios> ConsultaPinUsuario(List<Usuarios> lista_usuario)...
```

```
public List<Usuarios> ModificarUsuario(List<Usuarios> lista_usuario)...
```

```
public List<Usuarios> MostrarUsuarios(List<Usuarios> lista_usuario)...
```

```
public List<Usuarios> CompararUsuario(List<Usuarios> lista_usuario)...
```

```
public List<Usuarios> EliminarUsuario(List<Usuarios> lista_usuario)...
```

```
public List<Usuarios> AgregarUsuario(List<Usuarios> lista_usuario)...
```

```
public List<Usuarios> CompararCedula(List<Usuarios> lista_usuario)...
```

```
public List<Usuarios> CompararCorreo(List<Usuarios> lista_usuario)...
```

```
public List<Usuarios> ObtenerHuellas(List<Usuarios> lista_usuario)...
```

```
public List<Usuarios> CompararPin(List<Usuarios> lista_usuario)...
```

```
public List<Logs> MostrarEventos(List<Logs> lista_eventos)...
```

```
public List<Logs> AgregarEvento(List<Logs> lista_eventos)...
```

```
public int ObtenerID_Usuario(int id)...
```

```
public int ObtenerID_Eventos(int id)...
```

Código 2.13. Consultas LINQ.

```

public List<Usuarios> AgregarUsuario(List<Usuarios> lista_usuario)
{
    Usuarios nuevoUsuario = new Usuarios(identificadorUsuario, cedu
    miBase.Usuarios.InsertOnSubmit(nuevoUsuario);
    miBase.SubmitChanges();
    var consulta = from usuarios in miBase.Usuarios
                  where usuarios.Id == identificadorUsuario
                  select new
                  {
                      ID = usuarios.Id,
                      CEDULA = usuarios.Cedula,
                      ALIAS = usuarios.Alias,
                      NOMBRE = usuarios.Nombre,
                      APELLIDO = usuarios.Apellido,
                      PASSWORD = usuarios.Password,
                      TIPO = usuarios.Tipo,
                      DEPARTAMENTO = usuarios.Departamento,
                      CORREO = usuarios.Correo,
                      PIN = usuarios.Pin,
                      HUELLA = usuarios.Huella,
                      ROSTRO = usuarios.Rostro
                  };
    foreach (var item in consulta)
    {
        Usuarios cargar = new Usuarios(item.ID, item.CEDULA, item.A
        lista_usuario.Add(cargar);
        usuaurioCRUD = lista_usuario;
    }
    return usuaurioCRUD;
}

```

Código 2.14. Agregar un usuario mediante LINQ.

2.2.5. CONFIGURACIÓN DE LOS TERMINALES DE ACCESO

Los terminales de acceso, teclado numérico y lector dactilar, no requieren configuración alguna ya que éstos son dispositivos tipo Plug And Play; estos dispositivos solo requieren ser detectados por la aplicación para poder ser usados.

Las cámaras IP no requieren configuración del dispositivo ya que éstas también se auto-configuran al ser detectadas por la aplicación; sin embargo, si el administrador o el sistema lo requiere se puede cambiar el protocolo, puerto, usuario y contraseña para la cámara, como se muestra en la Figura 2.53.

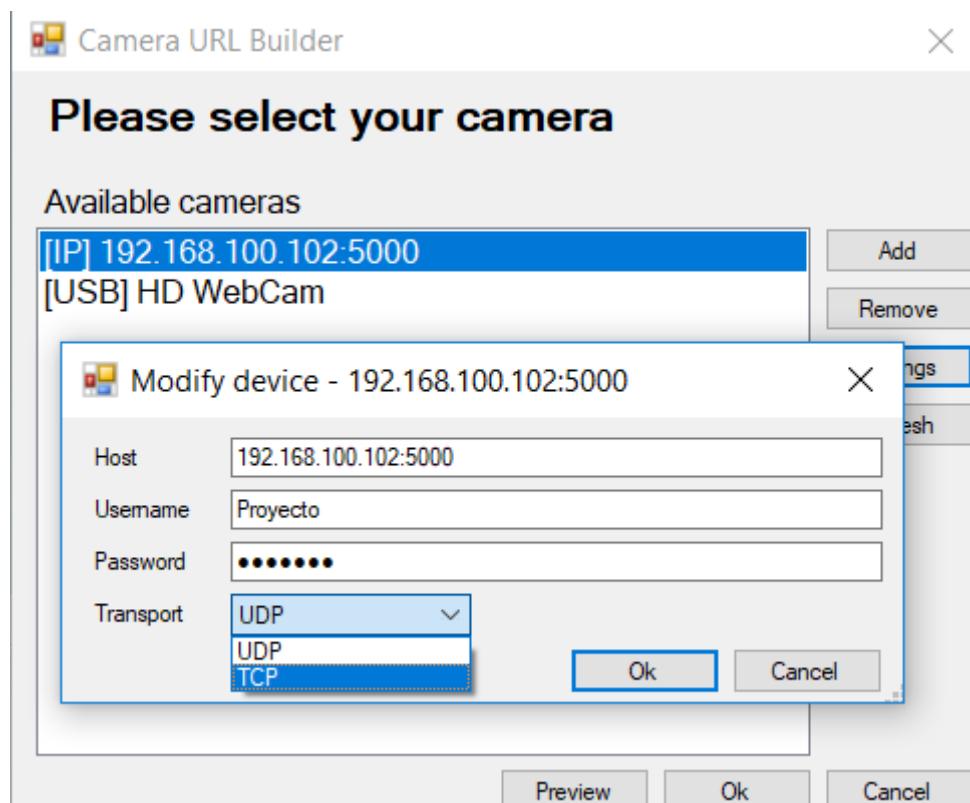


Figura 2.53. Configuración de la cámara de video.

2.2.6. CONFIGURACIONES DE EQUIPOS INTERMEDIOS

Al igual que los terminales de acceso los equipos intermedios no requieren configuración, al menos que el administrador lo desee; para el desarrollo de este proyecto se utilizó un Router Cable/DSL TL-R460 de 4 puertos LAN 10/100Mbps y 1 puerto WAN, el mismo que tiene habilitado el DHCP para asignar dinámicamente las direcciones IP a los dispositivos que se encuentran conectados, en este caso para la computadora donde se encuentra alojada la aplicación y para la Cámara IP que se encargará de tomar y enviar las imágenes a la aplicación de escritorio.

En la Figura 2.54 se muestra la configuración realizada en el router a utilizar, se puede observar el rango de direcciones y los dispositivos conectados al router con su respectiva dirección IP asignada de manera dinámica.

DHCP Settings

DHCP Server: Disable Enable

Start IP Address: 192.168.100.100

End IP Address: 192.168.100.199

Address Lease Time: 120 minutes (1~2880 minutes, the default value is 120)

Default Gateway: 0.0.0.0 (optional)

Default Domain:

Primary DNS: 0.0.0.0 (optional)

Secondary DNS: 0.0.0.0 (optional)

Save

DHCP Clients List

ID	Client Name	MAC Address	Assigned IP	Lease Time
1	none	4C-B0-08-EA-F0-40	192.168.100.102	01:58:53
2	ChristianArroyo	98-29-A6-32-B0-F9	192.168.100.101	01:40:50

Refresh

Figura 2.54. Configuración del router.

La placa Arduino no necesita configuración, pero sí de programación; ya que esta se encargará de comunicar la aplicación de escritorio con las cerraduras magnéticas y permite escoger la sala del Laboratorio de Redes a la cual se desea ingresar. El Código 2.15 se muestra parte de la programación realizada en el Arduino.

```
void setup()
{
    Serial.begin(9600);
    pinMode(Cerradura_1, OUTPUT);
    pinMode(Cerradura_2, OUTPUT);
    pinMode(Led_Sala_1, OUTPUT);
    pinMode(Led_Sala_2, OUTPUT);
    pinMode(Rostro, OUTPUT);
    pinMode(Huella, OUTPUT);
    pinMode(Sala_1, INPUT);
    pinMode(Sala_2, INPUT);
    pinMode(Pin, OUTPUT);
    digitalWrite(Led_Sala_1, LOW);
    digitalWrite(Led_Sala_2, LOW);
}

void loop()
{
    if (Serial.available() > 0)
    {
        int abrir = Serial.read();
        if (abrir == '1')
        {
            digitalWrite(Cerradura_1, HIGH);
            delay(3000);
            digitalWrite(Cerradura_1, LOW);
        }
    }
}
```

Código 2.15. Programación del Arduino.

2.2.7. UBICACIÓN FÍSICA DE LOS TERMINALES DE ACCESO

Los terminales de acceso se pueden ubicar en cualquier lugar al cual se tenga acceso fácil y directo. Para el desarrollo de este proyecto se colocó los terminales de acceso en la pared cercano a la puerta de entrada del “Laboratorio de Comunicaciones Digitales” que se encuentra cercano al Laboratorio de Redes como se muestra en la Figura 2.56.

La ventaja en colocar los periféricos en la pared son la de tener acceso directo a los equipos todo el tiempo, es decir aporta una mayor disponibilidad; aun así, las desventajas son mayores ya que al estar los dispositivos en la pared estos están expuestos al mal uso por parte de los usuarios no autorizados; adicionalmente, se necesita de extensiones eléctricas, extensiones USB y canaletas para poder conectar los dispositivos periféricos a la red eléctrica y al computador donde se encuentra la aplicación de escritorio, incrementando los costos de implementación.



Figura 2.55. Ubicación física de los terminales de acceso en la pared.

En la Figura 2.56 se muestra una vista externa de la “Caja Electrónica” la cual contiene los terminales de acceso (teclado numérico y lector dactilar), los equipos intermedios (router, Arduino y usb hub) y distintos componentes eléctricos y electrónicos (luces led, resistencias, relés, cables de conexión, extensión eléctrica y transformadores). En la Figura 2.57 se observa una vista interna de la “Caja Electrónica”.



Figura 2.56. Vista externa caja electrónica.

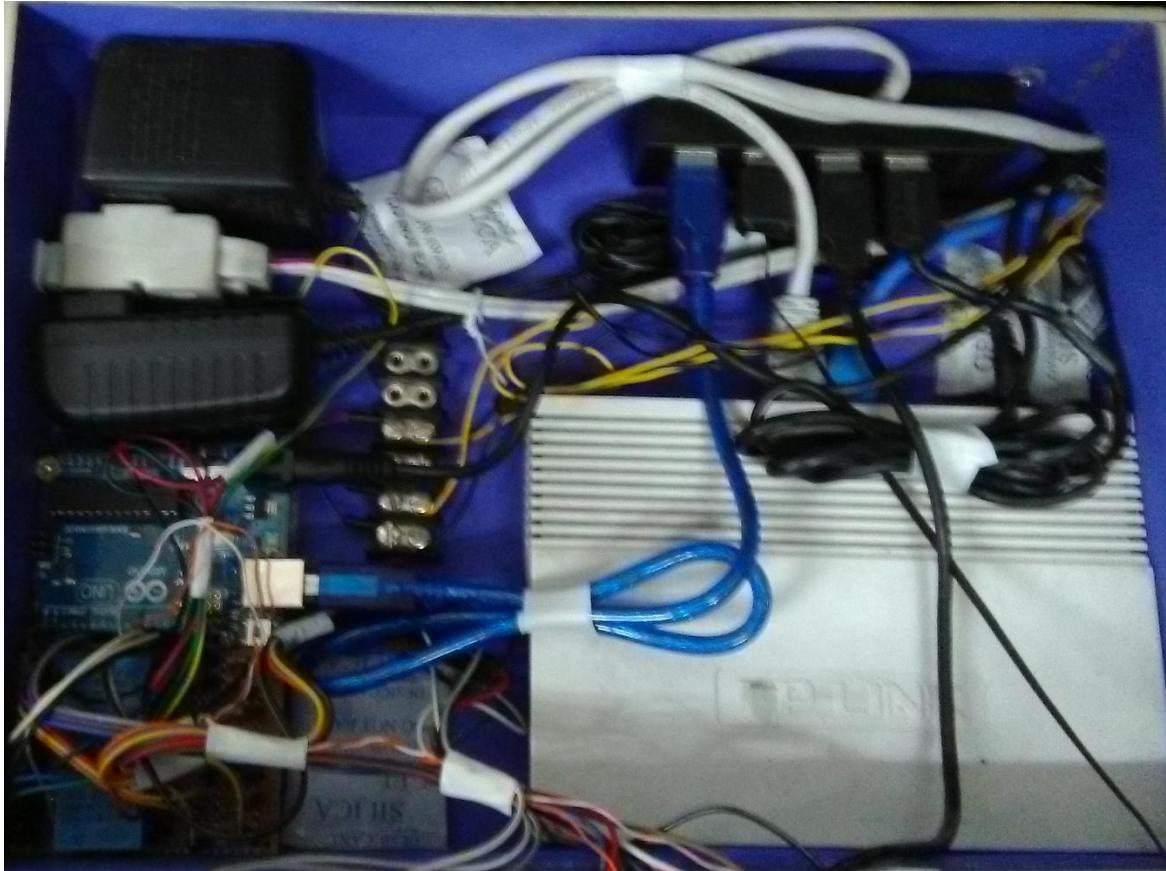


Figura 2.57. Vista interna caja electrónica.

2.2.8. CABLEADO PARA LA COMUNICACIÓN ENTRE LOS EQUIPOS

Para la comunicación entre los dispositivos periféricos y la aplicación de escritorio es necesario uso de cables UTP con interfaz física RJ45 de la categoría 5e con la norma EIA/TIA-568-B para intercomunicar la cámara IP con el router y este a la vez con el computador donde se encuentra la aplicación de escritorio. Adicionalmente, es necesario el uso de un cable de extensión USB 2.0 activo.

Finalmente, se necesita usar extensiones eléctricas para conectar los dispositivos periféricos que se conectan directamente a la corriente eléctrica, como lo son las cámaras IP. En Figura 2.58 se muestra el esquema del cableado para la comunicación entre equipos de los dos laboratorios.

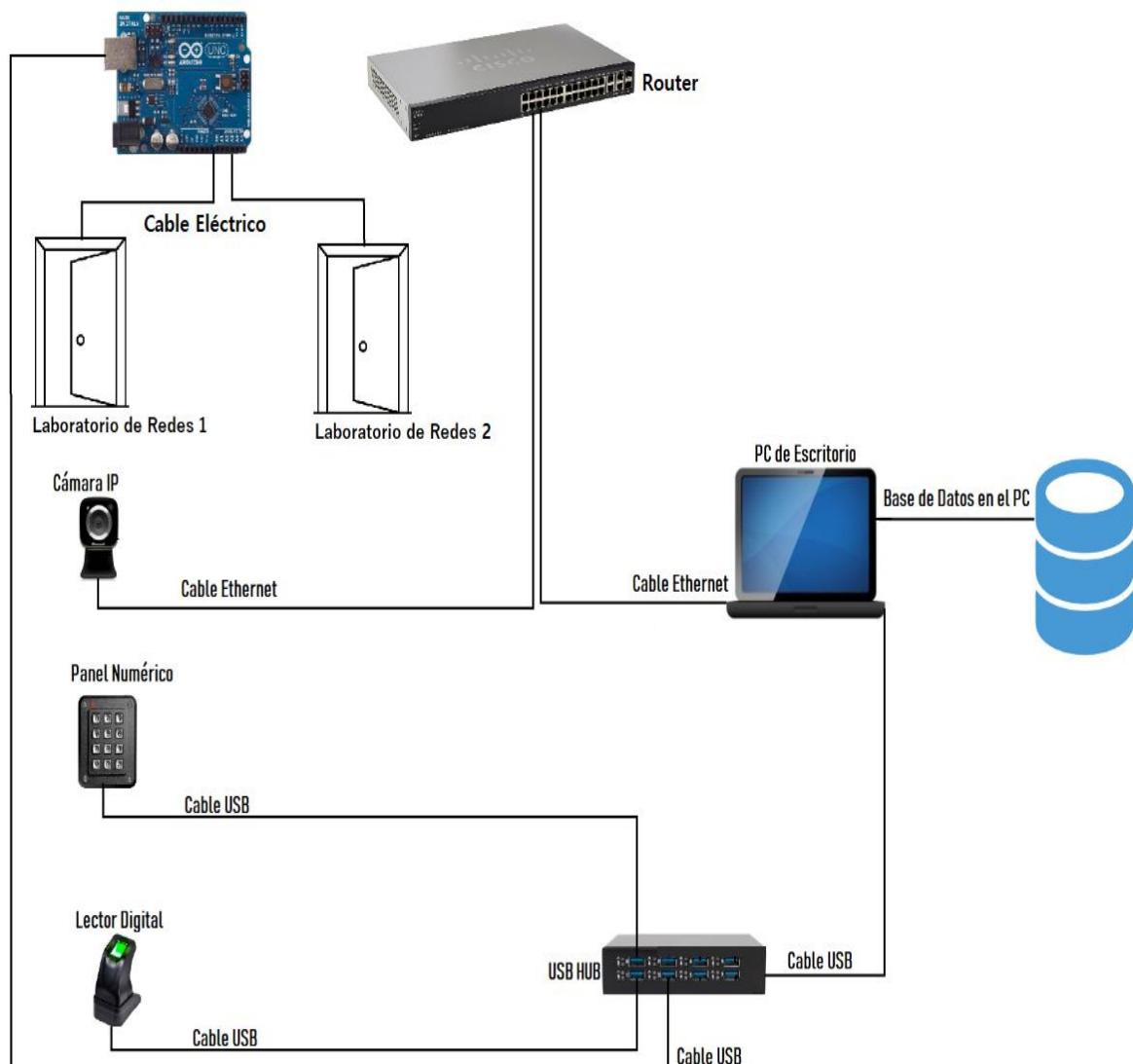


Figura 2.58. Implementación del sistema de seguridad.

En las Figuras 2.59 a 2.62 se muestra la cerradura y el cableado del sistema prototipo de seguridad realizado en una de las salas del Laboratorio de Redes.



Figura 2.59. Interruptor para abrir la cerradura internamente.



Figura 2.60. Cableado hacia el techo raso.



Figura 2.61. Cerradura en la puerta.



Figura 2.62. Acercamiento a la cerradura.

En la Tabla 2.7 se muestra la tercera actualización del tablero Kanban, en ella se muestra que las tareas que se encontraban en proceso han finalizado y nuevas tareas ocupan su posición. Para este punto, las tareas que se encuentran en proceso corresponden a la fase de resultados del sistema prototipo de seguridad.

Tabla 2.7. Tercera actualización tablero Kanban.

TABLERO KANBAN		
TAREAS POR REALIZAR	TAREAS EN PROCESO	TAREAS FINALIZADAS
	Comprobaciones de funcionamiento de los módulos	Selección metodología de Desarrollo Software.
	Correcciones de los módulos.	Ánálisis situación actual laboratorios de la FIEE.
	Verificar la existencia de errores producidos al interconectar los equipos.	Encuesta de requerimientos.
	Depurar los errores producidos al interconectar los equipos.	Diagramas de casos de uso.
	Probar el funcionamiento de las cámaras, lectores de huellas y pines de acceso numérico.	Diagramas de secuencia.
	Se realizarán las encuestas de satisfacción.	Diagramas de actividades.
		Diagramas de clase.
		Diagramas entidad relación.
		Requerimientos físicos del sistema de seguridad.
		Instalación Visual Studio 2017.
		Instalación Librerías Ozeki, EmguCV, OpenCV, ZKTeco e iText.
		Creación de los formularios de la interfaz gráfica.
		Implementación de cada módulo.
		Configuración de los terminales de acceso.
		Configuraciones de equipos intermedios.
		Ubicación física de los terminales de acceso.
		Extensión del cableado necesario para la comunicación entre los equipos.

3. RESULTADOS Y DISCUSIÓN

En este capítulo se describen las pruebas realizadas al sistema prototipo de seguridad, las mismas que pretenden verificar el cumplimiento de los requisitos funcionales y no funcionales y la de descubrir posibles errores en la ejecución de la aplicación de escritorio.

Las pruebas se centrarán, principalmente, en el funcionamiento correcto de la aplicación, y así verificar que los dispositivos periféricos cumplen con su función asignada. Para verificar el correcto funcionamiento de la aplicación se procederá a realizar los siguientes pasos:

- Se harán las comprobaciones de funcionamiento de los módulos.
- Se realizarán correcciones de los módulos.
- Verificar la existencia de errores producidos al interconectar los equipos.
- Depurar los errores producidos al interconectar los equipos.
- Probar el funcionamiento de las cámaras, lectores de huellas y pines de acceso numérico.
- Se realizarán las encuestas de satisfacción.

3.1. COMPROBACIÓN Y CORRECCIÓN DEL FUNCIONAMIENTO DE LOS MÓDULOS

Como ya se ha mencionado anteriormente los módulos involucrados en la aplicación de escritorio son: “Interfaz de Usuario”, “Lógica de Negocios”, “Acceso a Datos” y “Base de Datos”. Para comprobar el funcionamiento del módulo interfaz gráfica se procederá a validar que los elementos como los “textbox” trabajen tal y como se los configuró. Por ejemplo, en el registro de usuarios existe un campo llamado “Cedula”, este TextBox debe solo aceptar caracteres numéricos y de lo contrario debe lanzar un mensaje de alerta, como se observa en la Figura 3.1. Cabe mencionar que todos los campos en el registro de usuario están validados, dependiendo del tipo de campo, para evitar que se ingresen datos inadecuados o incorrectos.

Al momento que el usuario administrador desee establecer conexión con una cámara, deberá primero escoger la “Cámara” con la cual desea establecer conexión; en la Figura 3.2 se puede observar la comprobación del funcionamiento al momento de tratar de establecer conexión a una cámara de video sin haber escogido una previamente.

Administracion Usuarios

CREAR NUEVA CUENTA.

ID: 5

CEDULA:

ALIAS:

NOMBRE:

APELLIDO:

CONTRASEÑA:

CONFIRMAR CONTRASEÑA:

TIPO:

DEPARTAMENTO:

CORREO:

PIN DE ACCESO:

HUELLA DACTILAR:

DETECCIÓN FACIAL:

Detección Facial Buscar Foto

Rostros Detectados: 0 Guardar Buscar Guardar

Advertencia X

Solo se permiten numeros.

Aceptar

Registrar Huella Verificar Huella

Informacion: 0 Serial Dispositivo: 0 Registro Huella:

Guardar Huella

TOMAR HUELLA

TOMAR FOTO BUSCAR FOTO

REGISTRAR ACTUALIZAR ELIMINAR REGRESAR

ID	CEDULA	ALIAS	NOMBRE	APELLIDO	PASSWORD	TIPO	DEPARTAME	CORREO	PIN	HUELLA	ROSTRO
1	1715959837	Reivax	Christian	Arroyo	123@Christi...	Administrador	DETRI	christian.arr...	421878	mspd14uaoj...	
2	1723509111	Sandrita	Sandra	Aguilar	Tnp8j452*	Administrador	DETRI	sandra.aguil...	2144	mspd147Oo...	

Figura 3.1. Validación de un textbox.

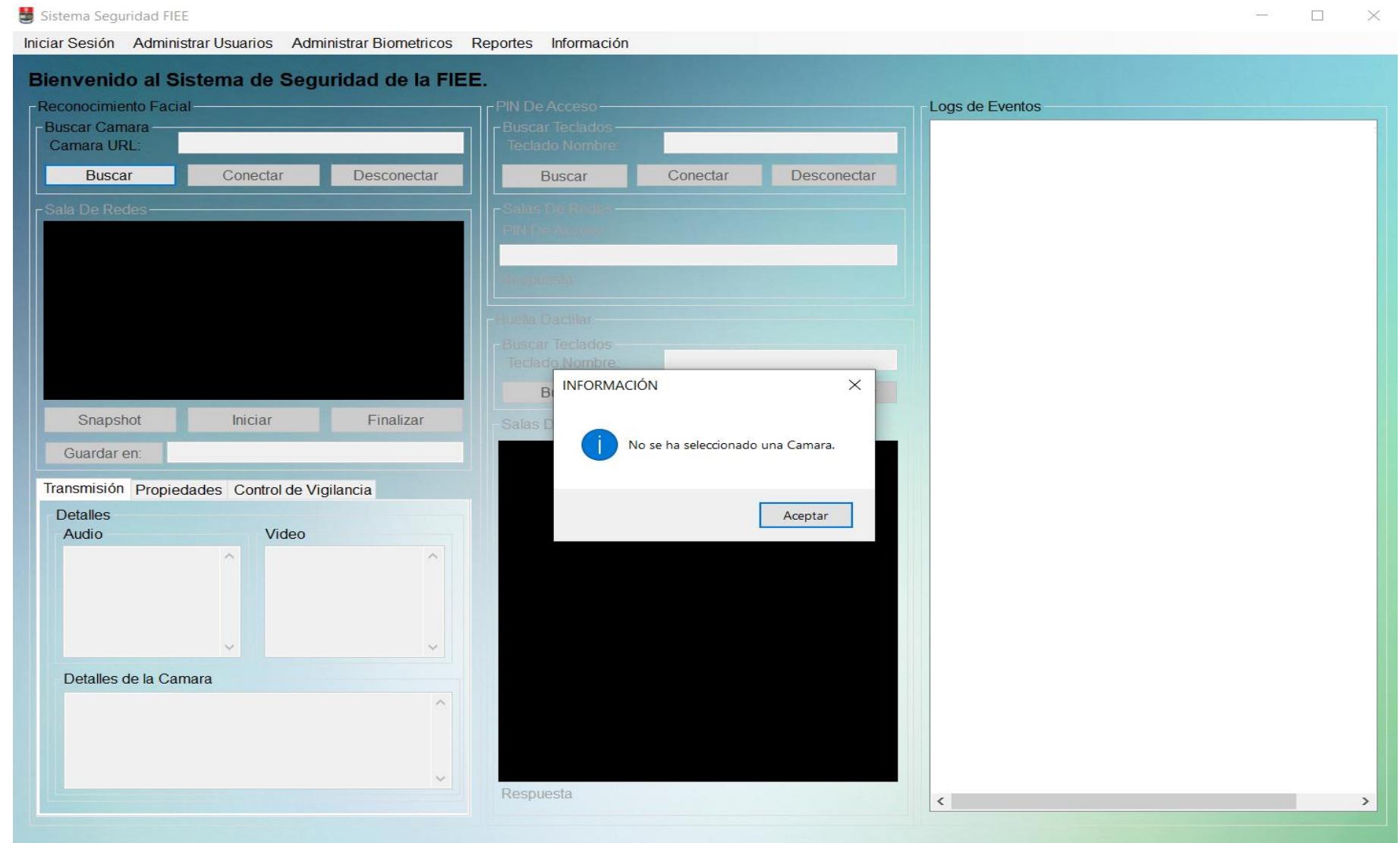


Figura 3.2. Validación selección de una cámara.

Para la “Lógica de Negocios”, “Acceso a Datos” y “Base de Datos” la mejor manera de comprobar la existencia de errores es realizando una consulta a la base de datos; ya que para ello el servidor, las interfaces y la base de datos deben estar activos. En este caso se procede a usar el evento de “Ver Logs”, debido que aquí permite consultar los eventos generados en la base de datos, filtrar la búsqueda e imprimir los eventos en un documento con el formato especificado.

En la Figura 3.3 se puede observar la consulta realizada a la base de datos con su respectiva búsqueda filtrada. En la Figura 3.4 se puede observar el documento impreso en el formato especificado.

A lo largo del desarrollo de este proyecto se produjeron varios errores, los mismos que fueron corregidos a medida que se iba programando. Los principales errores suscitados en el desarrollo del proyecto se mencionan a continuación:

- Incompatibilidad con la resolución de la pantalla: como se observa en la Figura 3.5 al momento de migrar la aplicación de un computador a otro se produce un problema en la adaptación a la resolución de la pantalla del nuevo computador, resultando en gráficos más grandes que no se ajustan a la pantalla. Para solucionar el problema, el administrador, debe modificar el tamaño de la fuente de los formularios mediante el uso de código en la clase de diseño de cada formulario, como se observa en la Figura 3.6.
- Base de datos incompatible: al migrar la aplicación de un computador a otro se produce un problema con la compatibilidad de la base de datos, como se observa en la Figura 3.7. El error se produce debido a la incompatibilidad de la versión de la base de datos con Microsoft Visual studio. Para solucionar el problema se debe instalar SQL Server en el computador y migrar la base de datos desde SQL Server. Una vez instalado SQL Server, en el nuevo computador sino se encuentra instalado, se debe copiar la base de datos de la aplicación en la ruta que por defecto SQL Server almacena las bases de datos, como se muestra en la Figura 3.8. Acto siguiente, se debe abrir SQL Server y agregar la base de datos a SQL Server, como se observa en la Figura 3.9. Una vez que se agregó la base de datos a SQL Server sobra por conectar la base de datos en el entorno de visual studio, como se observa en la Figura 3.10. Finalmente, se debe cambiar la cadena de conexión en la clase “MiDB” de la aplicación, como se observa en la Figura 3.11.
- Instaladores: es necesario instalar las librerías del lector dactilar, para que sea reconocido por el computador, como se expone en el capítulo 2.2.2.

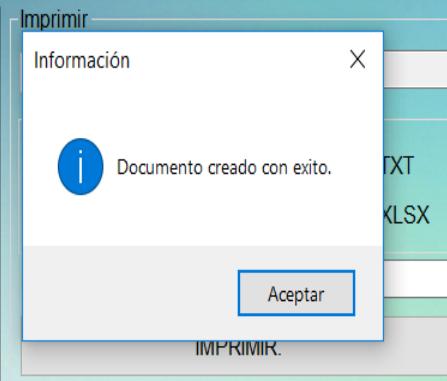
VER LOGS DE EVENTOS.

ID_LOGS	FECHA	EVENTO
1	8/5/2019 4:20:25	El Administrador Christian Arroyo ha iniciado sesion.
2	8/5/2019 4:25:27	El Administrador Christian Arroyo ha iniciado sesion.
3	8/5/2019 4:25:37	Conectando a Camara USB.
4	8/5/2019 4:25:37	Estado de la Camara USB Sala De Redes 1: Streaming.
5	8/5/2019 4:25:37	Se ha conectado a la camara HD WebCam Sala De Redes 1.
6	9/5/2019 2:52:04	El Administrador Christian Arroyo ha iniciado sesion.
7	9/5/2019 2:52:58	Se ha conectado al teclado Standard PS/2 Keyboard Sala De Redes 1.
8	9/5/2019 2:53:01	El usuario Christian Arroyo ha ingresado a la Sala De Redes 1.
9	9/5/2019 2:53:23	El usuario Christian Arroyo ha ingresado a la Sala De Redes 1.
10	11/5/2019 2:14:10	El Administrador Pablo Hidalgo ha iniciado sesion.
11	11/5/2019 2:14:20	Se ha conectado el teclado Standard PS/2 Keyboard Sala De Redes 1

124

BUSCAR LOGS DE EVENTOS.

ID_LOGS	FECHA	EVENTO
1	8/5/2019 4:20:25	El Administrador Christian Arroyo ha iniciado sesion.
2	8/5/2019 4:25:27	El Administrador Christian Arroyo ha iniciado sesion.
6	9/5/2019 2:52:04	El Administrador Christian Arroyo ha iniciado sesion.
8	9/5/2019 2:53:01	El usuario Christian Arroyo ha ingresado a la Sala De Redes 1.
9	9/5/2019 2:53:23	El usuario Christian Arroyo ha ingresado a la Sala De Redes 1.
14	11/5/2019 2:33:43	El Administrador Christian Arroyo ha iniciado sesion.
15	12/5/2019 14:43:....	El Administrador Christian Arroyo ha iniciado sesion.
31	13/5/2019 1:08:2...	El Administrador Christian Arroyo ha iniciado sesion.



Imprimir

Buscar por: Arroyo

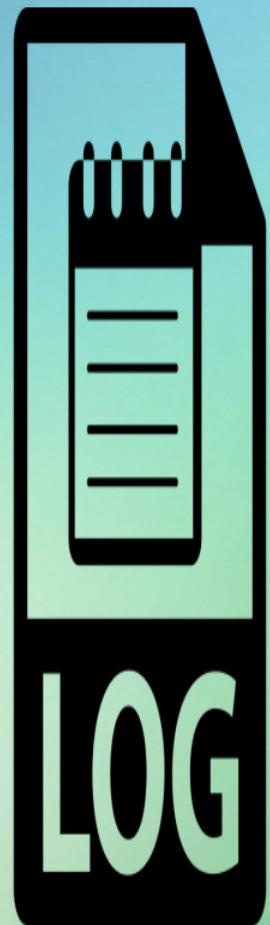
Guardar en: C:\Users\Christian Arroyo\Desktop

Formato

PDF TXT
 DOC XLSX

Nombre Documento: Prueba

IMPRIMIR.



REGRESAR.

Figura 3.3. Acceso y búsqueda en la datos y base de datos.

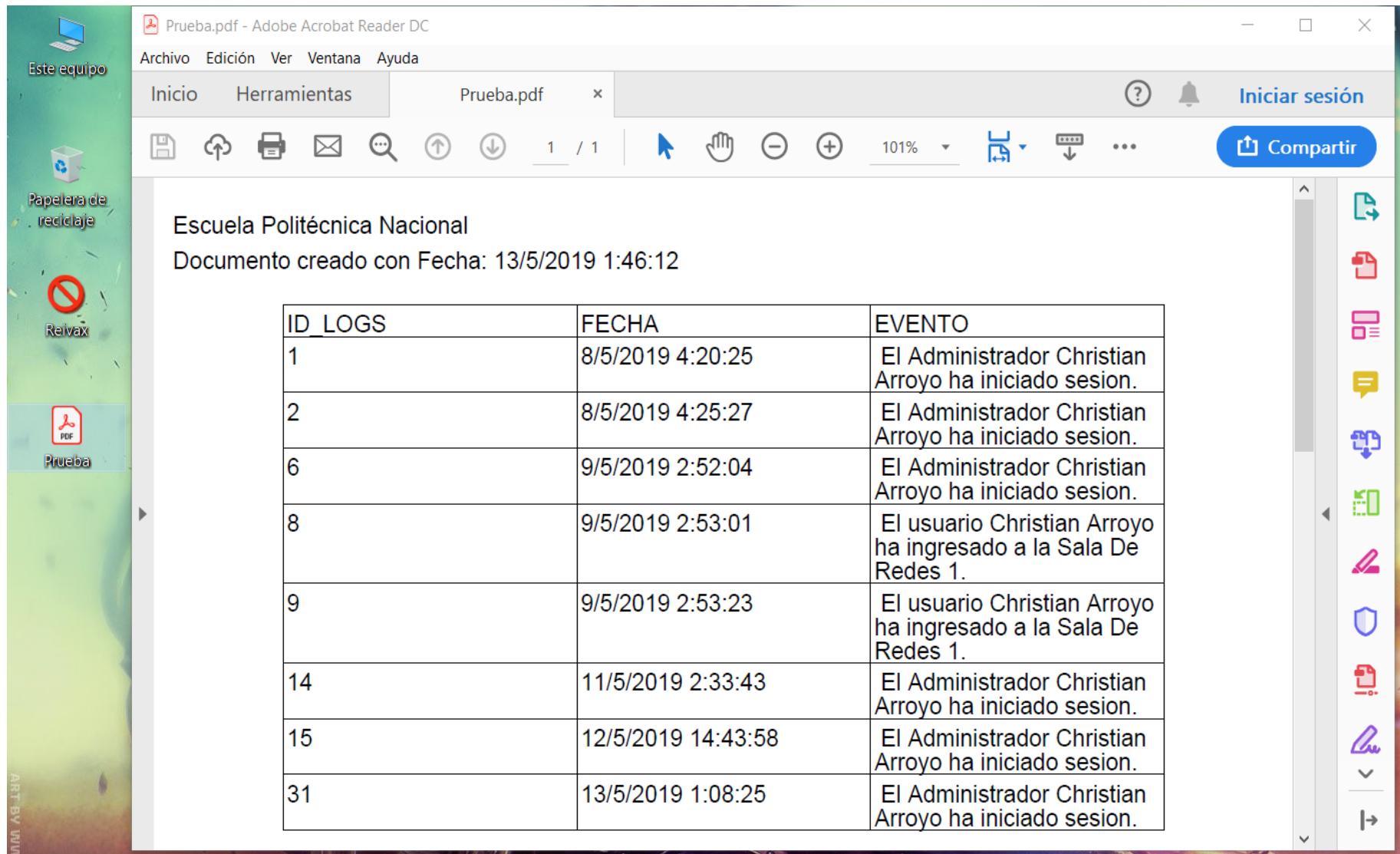


Figura 3.4. Documento impreso en el formato especificado.

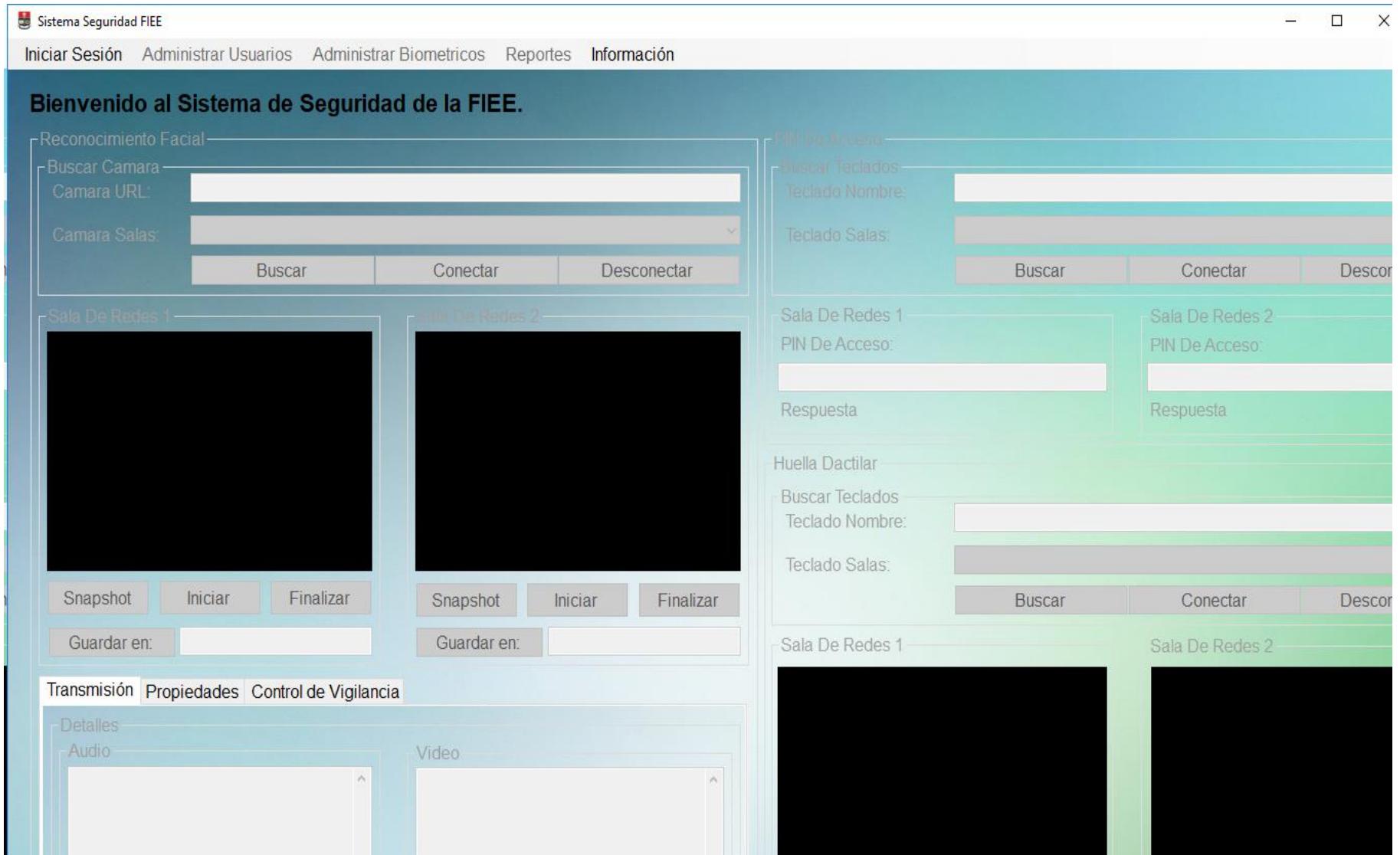


Figura 3.5. Incompatibilidad con la resolución de pantalla.

127

The screenshot shows the Visual Studio IDE interface. The main window displays the code for 'Sistema_Seguridad_FIEE.cs [Diseño]'. The code is a constructor for a Windows Form named 'Cliente'. It sets various properties like AutoScaleDimensions, AutoSizeMode, BackColor, BackgroundImage, ClientSize, Controls, Font, FormBorderStyle, Icon, MainMenuStrip, Name,StartPosition, Text, and FormClosing and Load event handlers. It also performs layout operations on several group boxes and tabs. The Solution Explorer on the right shows the project structure with files like 'Properties', 'Referencias', 'Evento', 'Librerias', 'Resources', 'Administracion_Usuarios.cs', 'Agregar_Lectores.cs', 'Agregar_Teclados.cs', 'Inicio_Sesion.cs', 'Logs_Eventos.cs', 'Sistema_Seguridad_FIEE.cs' (selected), 'Sistema_Seguridad_FIEE.Designer.cs', and 'Sistema_Seguridad_FIEE.resx'. The status bar at the bottom indicates the code is 90% complete, with line 1611, column 13, character 13, and 'INS' mode.

```
// Sistema_Seguridad_FIEE
//
this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 12F);
this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
this.BackColor = System.Drawing.SystemColors.ActiveCaption;
this.BackgroundImage = ((System.Drawing.Image)(resources.GetObject("$this.BackgroundImage")));
this.BackgroundImageLayout = System.Windows.Forms.ImageLayout.Stretch;
this.ClientSize = new System.Drawing.Size(909, 499);
this.Controls.Add(this.groupBox_Huella);
this.Controls.Add(this.groupBox_Pin);
this.Controls.Add(this.groupBox_Logs);
this.Controls.Add(this.groupBox_Facial);
this.Controls.Add(this.etiquetaBienvenida);
this.Controls.Add(this.menuStrip_Administracion);
this.DoubleBuffered = true;
this.Font = new System.Drawing.Font("Microsoft Sans Serif", 7F);
this.FormBorderStyle = System.Windows.Forms.FormBorderStyle.FixedSingle;
this.Icon = ((System.Drawing.Icon)(resources.GetObject("$this.Icon")));
this.MainMenuStrip = this.menuStrip_Administracion;
this.Name = "Sistema_Seguridad_FIEE";
this.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen;
this.Text = "Sistema Seguridad FIEE";
this.FormClosing += new System.Windows.Forms.FormClosingEventHandler(this.Sistema_Seguridad_FIEE_FormClosing);
this.Load += new System.EventHandler(this.Sistema_Seguridad_FIEE_Load);
this.menuStrip_Administracion.ResumeLayout(false);
this.menuStrip_Administracion.PerformLayout();
this.groupBox_Facial.ResumeLayout(false);
this.groupBox_Sala2.ResumeLayout(false);
this.groupBox_Sala2.PerformLayout();
this.tabControl_Detalles.ResumeLayout(false);
this.tabPage1.ResumeLayout(false);
this.groupBox_Detalles.ResumeLayout(false);
this.groupBox_Video.ResumeLayout(false);
this.groupBox_Video.PerformLayout();
this.groupBox_Video.PerformLayout();
```

Figura 3.6. Corrección de incompatibilidad con la resolución de pantalla.

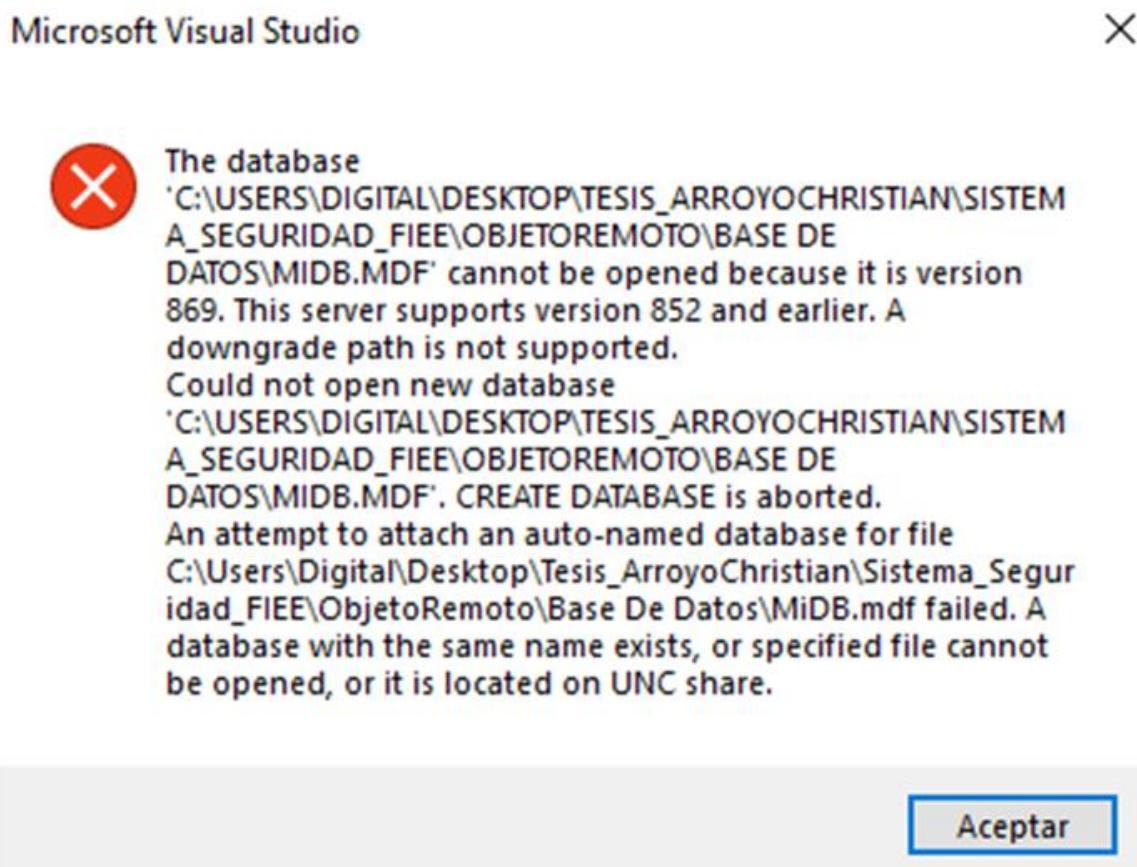


Figura 3.7. Base de datos incompatible.

equipo > Disco local (C:) > Archivos de programa > Microsoft SQL Server > MSSQL14.SQLEXPRESS > MSSQL > DATA			
Nombre	Fecha de modifica...	Tipo	Tamaño
master	13/6/2019 20:13	SQL Server Database Primary Data File	5.504 KB
mastlog	13/6/2019 20:13	SQL Server Database Transaction Log File	2.048 KB
MiDB	13/6/2019 21:58	SQL Server Database Primary Data File	8.192 KB
MiDB_log	13/6/2019 21:58	SQL Server Database Transaction Log File	8.192 KB
model	13/6/2019 20:13	SQL Server Database Primary Data File	8.192 KB
modellog	13/6/2019 20:13	SQL Server Database Transaction Log File	8.192 KB
MS_AgentSigningCertificate	30/5/2019 12:59	Certificado de seguridad	1 KB
MSDBData	13/6/2019 20:13	SQL Server Database Primary Data File	18.240 KB
MSDBLog	13/6/2019 20:13	SQL Server Database Transaction Log File	29.504 KB
tempdb	13/6/2019 20:16	SQL Server Database Primary Data File	8.192 KB
templog	13/6/2019 20:16	SQL Server Database Transaction Log File	8.192 KB

Figura 3.8. Bases de datos de SQL Server.

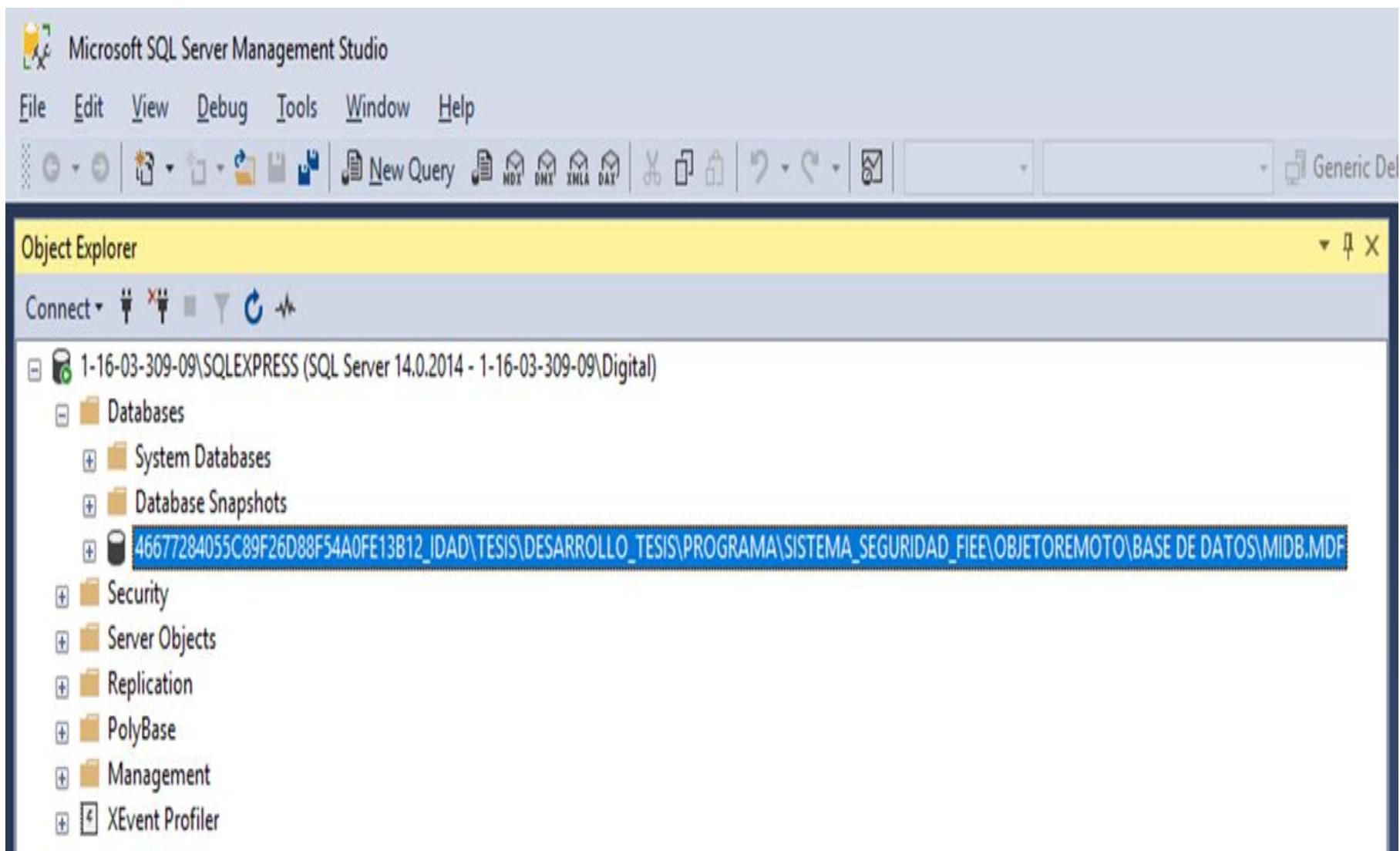


Figura 3.9. Migrando la base de datos del proyecto a SQL Server.

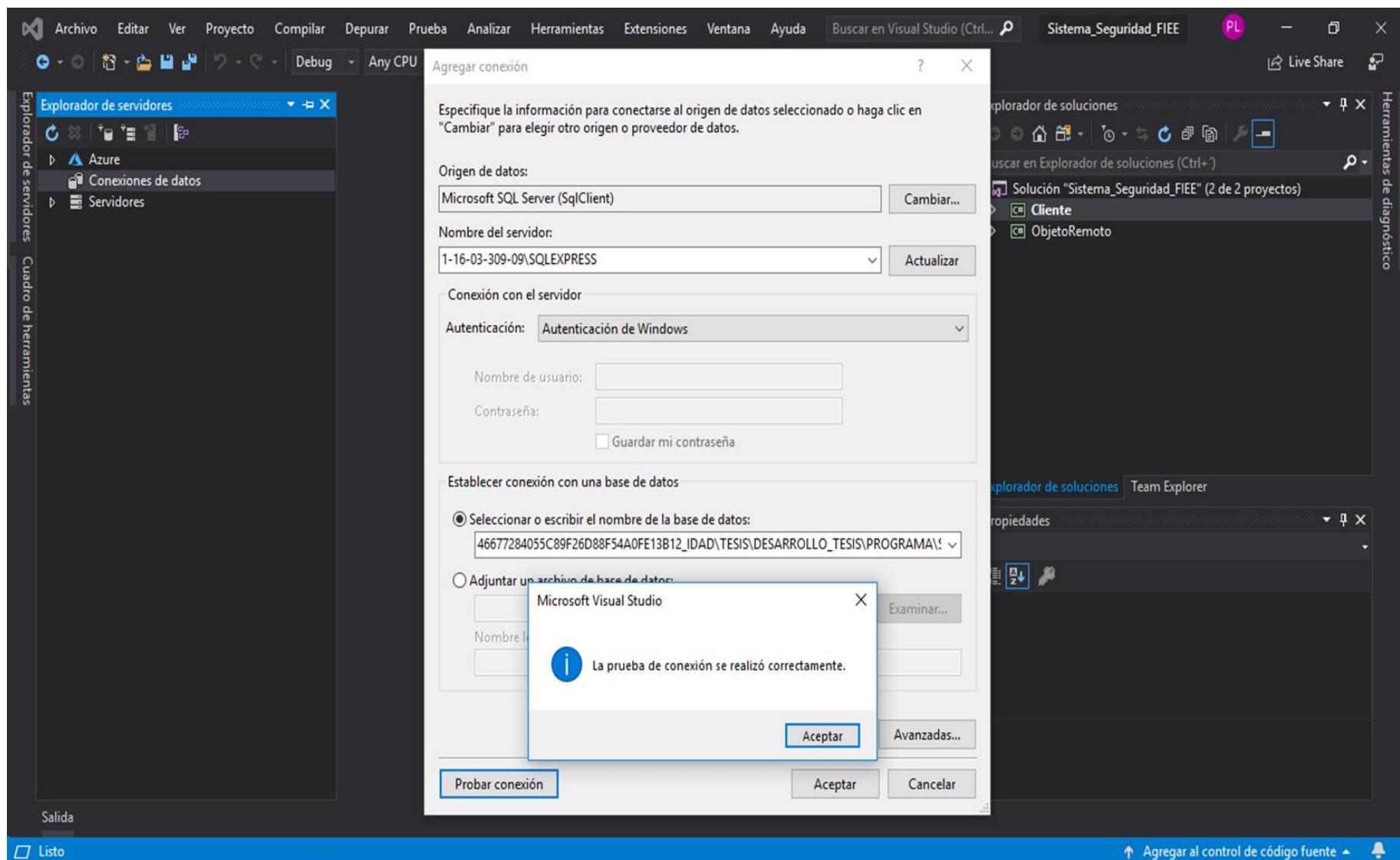


Figura 3.10. Conexión de la base de datos en el nuevo computador.

The screenshot shows the Visual Studio IDE interface. The main window displays the code for `MiDB.cs` in the `ObjetoRemoto` namespace. The code defines a `MiDB` class that inherits from `DataContext`, containing properties for `Usuarios` and `Logs`. The connection string is set to `Data Source=1-16-03-309-09\SQLEXPRESS;Initial Catalog=46677284055C89F26D88F54A0FE13812_IDAD\TESIS\0`.

```
using System.Collections.Generic;
using System.Data.Linq.Mapping;
using System.Threading.Tasks;
using System.Data.Linq;
using System.Linq;
using System.Text;
using System;

namespace ObjetoRemoto
{
    public class MiDB : DataContext
    {
        public Table<Usuarios> Usuarios;
        public Table<Logs> Logs;
        public MiDB() : base(@"Data Source=1-16-03-309-09\SQLEXPRESS;Initial Catalog=46677284055C89F26D88F54A0FE13812_IDAD\TESIS\0")
    }
}
```

The Solution Explorer on the right shows the project structure for "Sistema_Seguridad_FIEE" with two projects: `Cliente` and `ObjetoRemoto`. The `ObjetoRemoto` project contains files for `Properties`, `Referencias`, `Base De Datos`, `Servicios`, and several code files including `App.config`, `IObjetoRemoto.cs`, `Logs.cs`, `MiDB.cs` (which is selected), `ObjetoRemoto.cs`, `Servidor.cs`, and `Usuarios.cs`.

Figura 3.11. Actualizando la cadena de conexión de la base de datos.

3.2. VERIFICAR Y DEPURAR LA EXISTENCIA DE ERRORES PRODUCIDOS AL INTERCONECTAR LOS EQUIPOS

La interconexión de los equipos depende simplemente del administrador y de cómo el administrador configure los equipos; a continuación, se enumera una lista con los posibles errores que se pueden producir al interconectar los equipos, por una mala configuración o por elementos defectuosos, y que son independientes de la aplicación:

- Dispositivos de acceso defectuosos.
- Cables de conexión de los dispositivos de acceso defectuosos.
- Cables de conexión USB o UTP defectuosos o incorrectos.
- Conector eléctrico de las cámaras o del Router defectuoso.
- USB Hub defectuoso.
- USB Hub con bajo nivel de voltaje (por debajo de los 5 voltios).
- Mala configuración del router en la asignación de direcciones IP a los dispositivos
- Placa Arduino mal conectada por lo cual no puede establecer comunicación con la aplicación de escritorio como se muestra en la Figura 3.12. Para evitar este error se debe comprobar que el puerto “COM” de computador y el cable de conexión del Arduino estén en buenas condiciones y que estén todos los controladores debidamente instalados y actualizados.

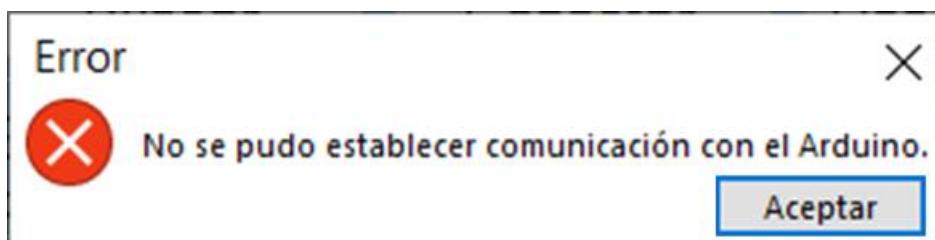


Figura 3.12. Fallo en la conexión con el Arduino.

3.3. FUNCIONAMIENTO DE LAS CÁMARAS, LECTORES DE HUELLAS Y PINES DE ACCESO NUMÉRICO

Para comprobar el funcionamiento de la cámara de video se hará uso del formulario principal, donde se escogerá una cámara que se encuentre conectada. En la Figura 3.13 se puede observar que la cámara se encuentra activa y detectando la presencia de un rostro al cual ya habrá identificado. Adicionalmente, se observa en el log de eventos que el usuario se encuentra registrado.

En las cámaras se puede dar un error, no de funcionamiento, al intentar conectarse a una cámara que no se encuentra dentro de la red establecida. En la Figura 3.6 se muestra el error cuando se trata de conectar a una cámara en una red inalcanzable. En el log de eventos se mostrará simplemente como cámara desconectada.

Hay que tener en consideración que para que exista una buena detección facial las condiciones de iluminación y altura de la cámara sean las correctas. Si la iluminación del ambiente es muy baja (inferior a 200 lux [7]) no podrá haber detección facial, como se observa en la Figura 3.15; de igual manera si la altura, distancia o el ángulo del rostro a la cámara no es la apropiada (altura: por encima o debajo de rostro, distancia: mayor a 1.20 metros y ángulo: inferior a 20° [7]) no habrá detección por lo cual no se podrá reconocer el rostro del usuario como se observa en la Figuras 3.16 y 3.17.

La comprobación del funcionamiento del lector dactilar se lo hará en el evento “Agregar Lectores”, este formulario permite reconocer el dispositivo que se encuentra conectado; adicionalmente proporciona la información del nombre del dispositivo y el registro del mismo. En la Figura 3.18 se puede observar el funcionamiento del lector dactilar.

En la Figura 3.19, se detalla la comprobación del funcionamiento del teclado numérico el cual se lo realiza en el evento “Agregar Teclados” y de igual manera que con los lectores dactilares en este formulario permite reconocer los teclados conectados y proporciona información útil como lo es el nombre del teclado y su registro.

En la Figura 3.20 se puede apreciar el funcionamiento en la interfaz principal de usuario, donde se observa el funcionamiento del reconocimiento facial, reconocimiento dactilar y el ingreso por PIN de acceso. Para mayor información sobre el funcionamiento de la aplicación, se puede revisar el manual de usuario en el “Anexo E”.

En la Tabla 3.1 se muestra las estadísticas obtenidas tras realizar las pruebas de funcionamiento (reconocimiento facial, huella dactilar, PIN de acceso y registro de usuario) a un grupo de 25 individuos, entre estudiantes y profesores, de la FIEE de la EPN.

Finalmente, en la Figura 3.21 se muestran todas las librerías que se utilizaron para el desarrollo del proyecto; estas librerías hay que importarlas al proyecto para que funcione correctamente.

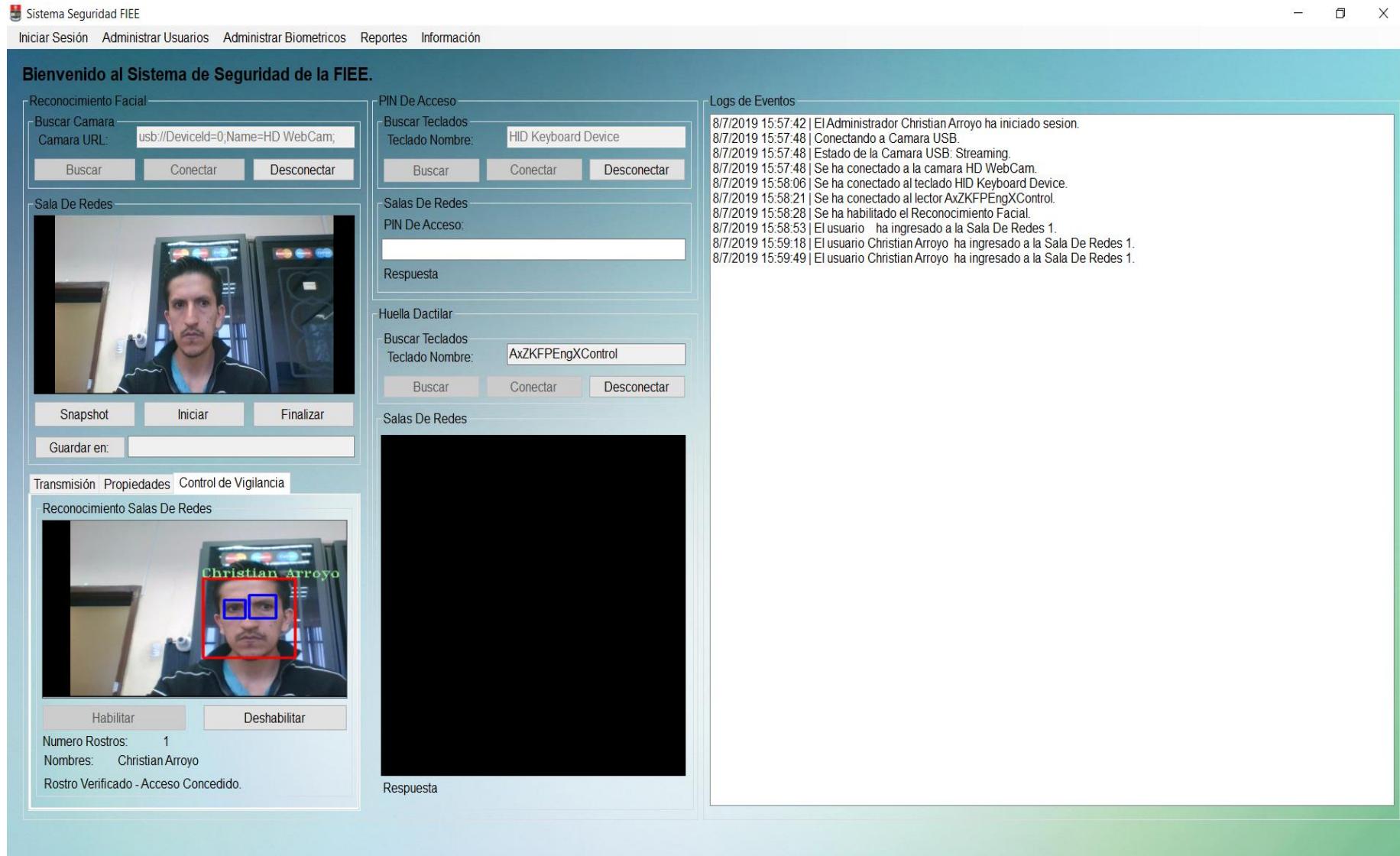


Figura 3.13. Comprobación del funcionamiento de la cámara.

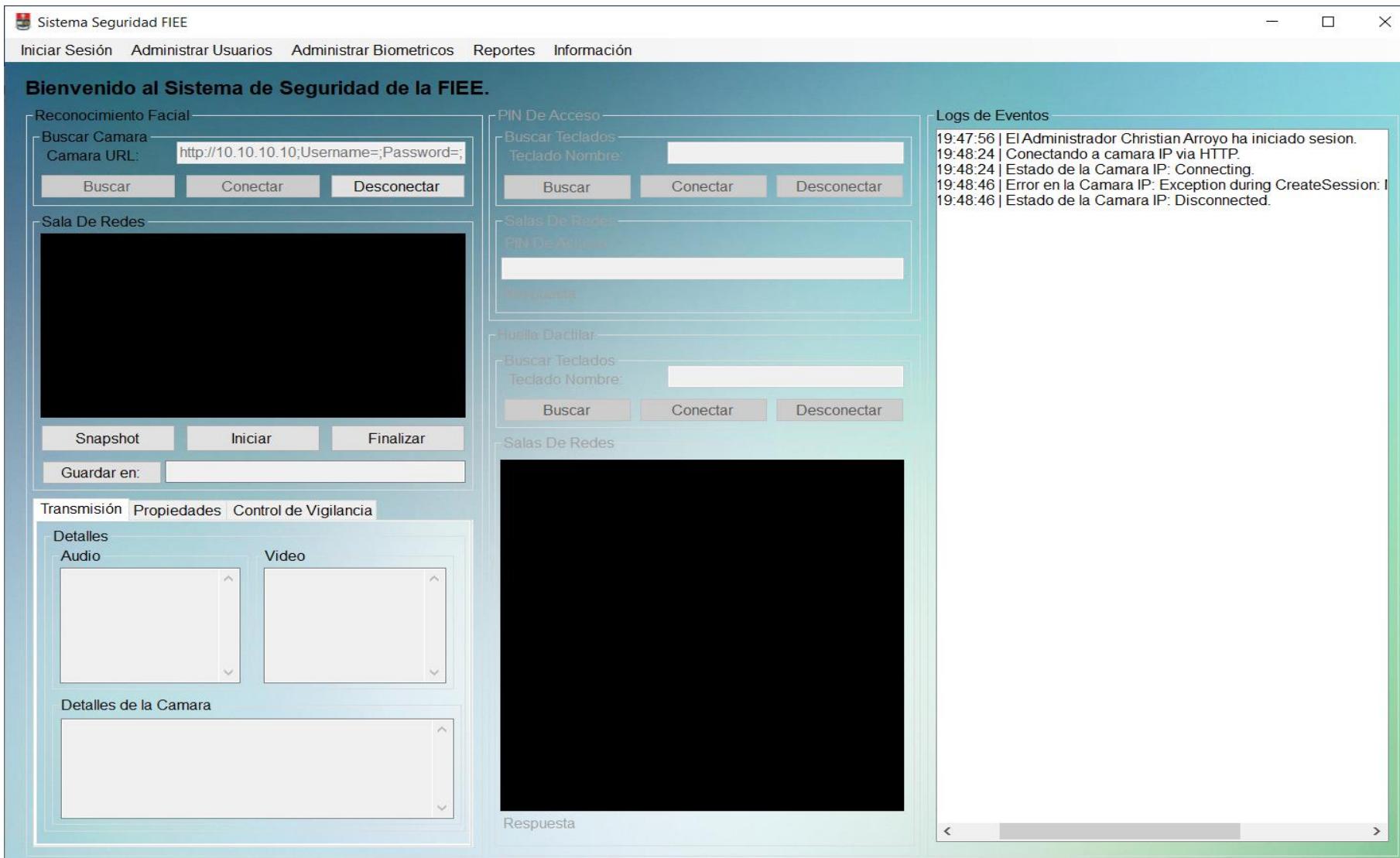


Figura 3.14. Cámara en red inalcanzable.



Figura 3.15. Imagen con malas condiciones de iluminación.



Figura 3.16. Imagen con malas condiciones de altura.



Figura 3.17. Imagen con mal ángulo de observación.

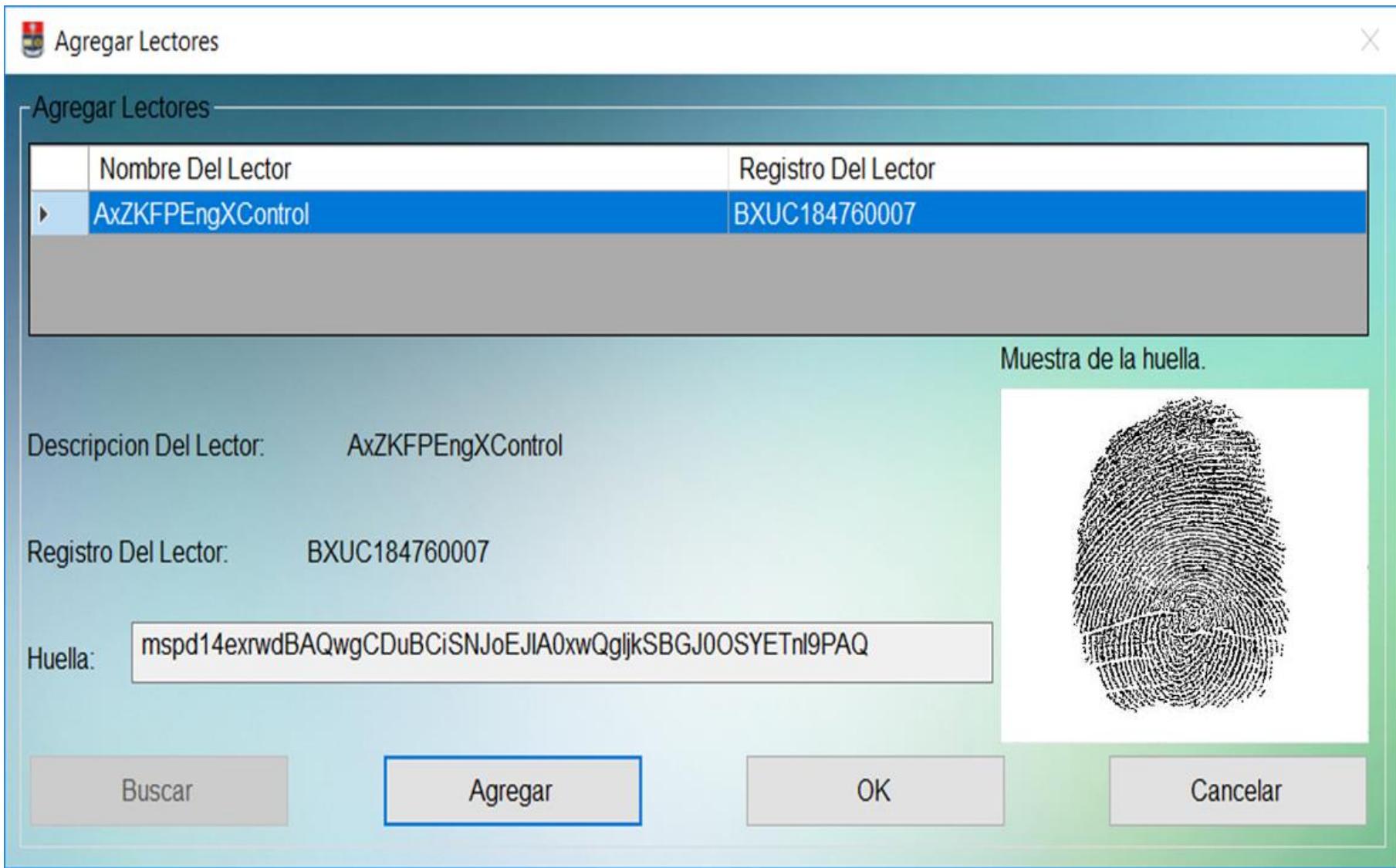


Figura 3.18. Funcionamiento del lector dactilar.

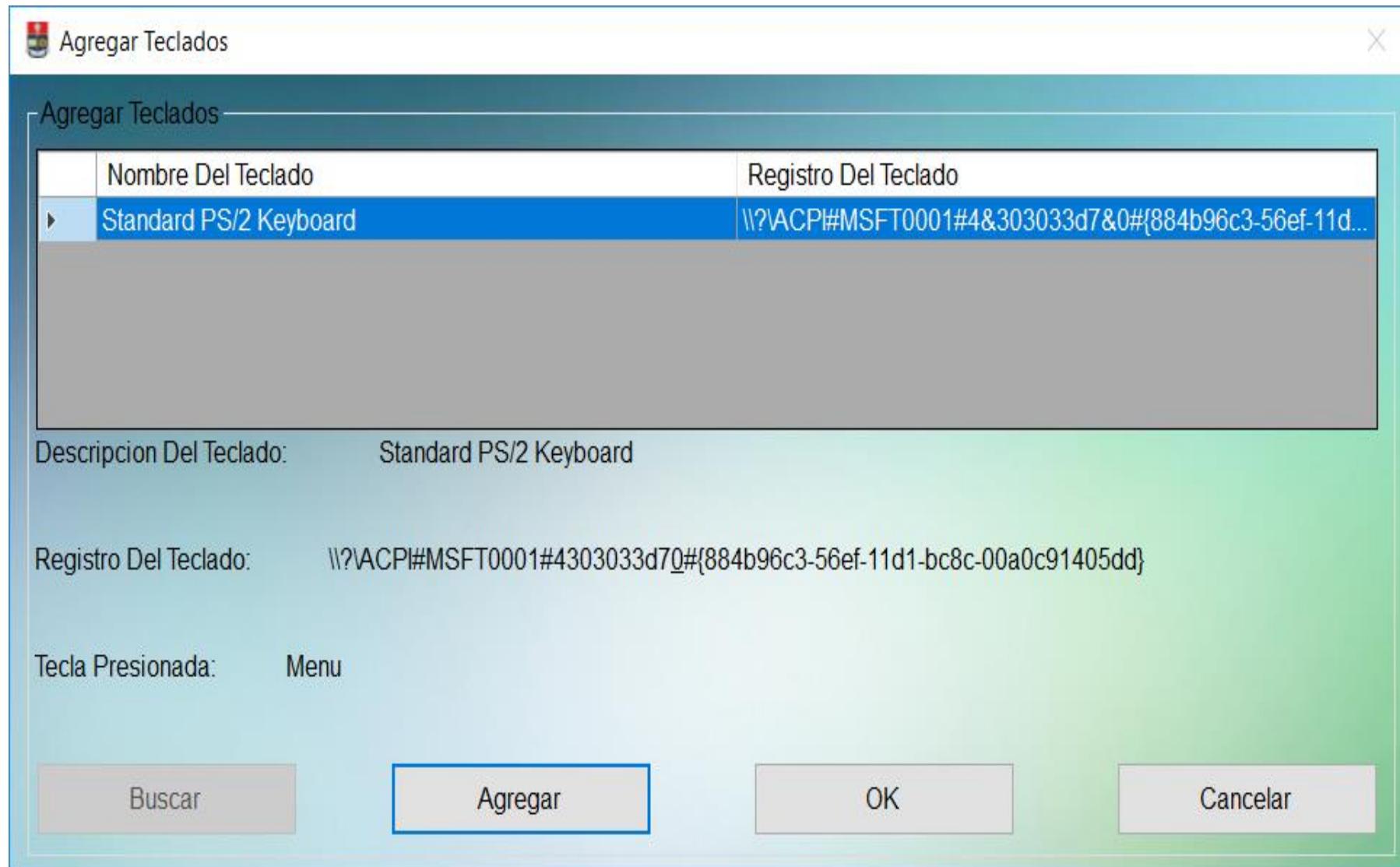


Figura 3.19. Funcionamiento del teclado numérico.

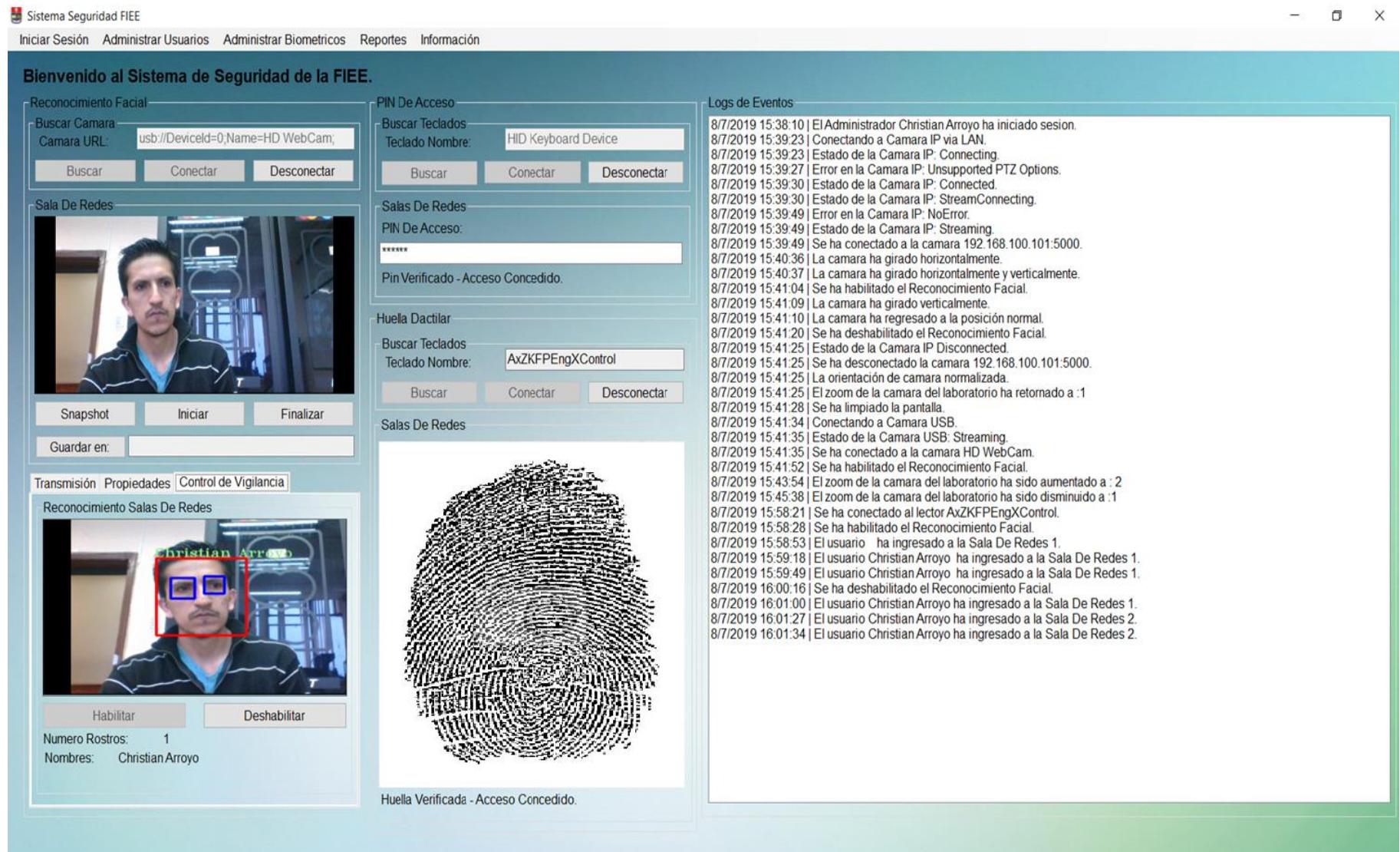


Figura 3.20. Resultados del funcionamiento de la aplicación.

Tabla 3.1. Estadísticas de las pruebas de funcionamiento.

ESTADÍSTICAS DE LAS PRUEBAS DE FUNCIONAMIENTO				
	RECONOCIMIENTO FACIAL	RECONOCIMIENTO DACTILAR	PIN DE ACCESO	REGISTRO DE USUARIO
Usuario 1	No	Si	Si	Si
Usuario 2	Si	Si	Si	Si
Usuario 3	Si	Si	Si	Si
Usuario 4	Si	Si	Si	Si
Usuario 5	Si	Si	Si	Si
Usuario 6	Si	Si	Si	Si
Usuario 7	Si	Si	Si	Si
Usuario 8	Si	Si	Si	Si
Usuario 9	Si	Si	Si	Si
Usuario 10	Si	No	Si	Si
Usuario 11	Si	Si	Si	Si
Usuario 12	No	Si	Si	No
Usuario 13	No	Si	Si	Si
Usuario 14	No	Si	Si	Si
Usuario 15	Si	Si	Si	Si
Usuario 16	Si	Si	Si	Si
Usuario 17	Si	Si	Si	Si
Usuario 18	No	Si	Si	Si
Usuario 19	Si	Si	Si	Si
Usuario 20	Si	Si	Si	Si
Usuario 21	No	No	Si	Si
Usuario 22	Si	Si	Si	Si
Usuario 23	Si	Si	Si	Si
Usuario 24	No	Si	Si	No
Usuario 25	No	No	Si	Si
Porcentaje de errores	32%	12%	0%	8%

 AxInterop.ZKFPEngXControl.dll	5/6/2018 0:34	Extensión de la ap...	36 KB
 cvextern.dll	9/4/2019 17:35	Extensión de la ap...	589 KB
 Emgu.CV.dll	9/4/2019 17:35	Extensión de la ap...	248 KB
 Emgu.CV.UI.dll	9/4/2019 17:35	Extensión de la ap...	124 KB
 Emgu.Util.dll	9/4/2019 17:35	Extensión de la ap...	32 KB
 haarcascade_eye	9/4/2019 17:35	Documento XML	495 KB
 haarcascade_frontalface_default	9/4/2019 17:35	Documento XML	1.226 KB
 Interop.ZKFPEngXControl.dll	5/6/2018 0:34	Extensión de la ap...	40 KB
 itextsharp.dll	8/5/2019 23:42	Extensión de la ap...	3.404 KB
 opencv_calib3d220.dll	9/4/2019 17:35	Extensión de la ap...	410 KB
 opencv_contrib220.dll	9/4/2019 17:35	Extensión de la ap...	442 KB
 opencv_core220.dll	9/4/2019 17:35	Extensión de la ap...	2.084 KB
 opencv_features2d220.dll	9/4/2019 17:35	Extensión de la ap...	796 KB
 opencv_flann220.dll	9/4/2019 17:35	Extensión de la ap...	30 KB
 opencv_gpu220.dll	9/4/2019 17:35	Extensión de la ap...	148 KB
 opencv_highgui220.dll	9/4/2019 17:35	Extensión de la ap...	866 KB
 opencv_imgproc220.dll	9/4/2019 17:35	Extensión de la ap...	1.405 KB
 opencv_legacy220.dll	9/4/2019 17:35	Extensión de la ap...	661 KB
 opencv_ml220.dll	9/4/2019 17:35	Extensión de la ap...	381 KB
 opencv_objdetect220.dll	9/4/2019 17:35	Extensión de la ap...	319 KB
 opencv_video220.dll	9/4/2019 17:35	Extensión de la ap...	233 KB
 stdole.dll	20/10/2018 19:15	Extensión de la ap...	32 KB

Figura 3.21. Librerías utilizadas en el proyecto.

3.4. ENCUESTAS DE SATISFACCIÓN

La encuesta de satisfacción, se la aplicó al personal de la FIEE de la EPN; la información sobre las encuestas se encuentra en el “Anexo F”. Los resultados de las encuestas realizadas se presentan, a continuación, en las Tablas 3.2 y 3.3.

Tabla 3.2. Resultados de las encuestas de satisfacción.

RESULTADOS ENCUESTA DE SATISFACCIÓN				
PREGUNTA	MUY DE ACUERDO	DE ACUERDO	POCO DE ACUERDO	NADA DE ACUERDO
¿La interfaz de usuario es amigable?	40%	60%	0%	0%
¿Aprender a manejar la aplicación de escritorio es fácil?	40%	60%	0%	0%
¿La información que se presenta es clara?	50%	50%	0%	0%
¿La información que se presenta, como mensajes de error y mensajes de información, es clara y entendible?	70%	30%	0%	0%
¿La organización de los menús y objetos visuales es lógica?	60%	40%	0%	0%
¿La aplicación de escritorio se ejecuta rápidamente?	50%	40%	10%	0%
¿La resolución de las imágenes obtenidas por las cámaras son de buena calidad?	60%	40%	0%	0%
¿Los parámetros como: zoom, brillo y contraste, establecidos para procesar las imágenes son buenos?	30%	70%	0%	0%
¿Las huellas obtenidas por los lectores dactilares son de buena calidad?	60%	40%	0%	0%
¿La sensibilidad del teclado es buena?	80%	20%	0%	0%
¿El tiempo de respuesta para las consultas es rápido?	70%	30%	0%	0%
¿Los formatos de impresión (PDF, WORD, Excel) de los eventos son adecuados?	50%	50%	0%	0%
¿La ubicación de los dispositivos de acceso es la adecuada?	60%	40%	0%	0%
¿Está de acuerdo con la calidad de los dispositivos de acceso?	30%	70%	0%	0%
¿Está de acuerdo con la calidad de las cerraduras?	60%	40%	0%	0%

Tabla 3.3. Resultados de las encuestas de satisfacción (continuación).

RESULTADOS ENCUESTA DE SATISFACCIÓN									
PREGUNTA	MUY DE ACUERDO		DE ACUERDO		POCO DE ACUERDO		NADA DE ACUERDO		
¿Está satisfecho/a con el producto final?	60%		40%		0%		0%		
¿Recomendaría el producto?	70%		30%		0%		0%		
¿Qué calificación le daría al producto?	10	9	8	7	6	5	4	3	2
	30%	30%	40%	0%	0%	0%	0%	0%	0%
¿Cuáles aspectos del software y hardware calificaría como malos o regulares?	Ninguno: 60%. Mejorar el Reconocimiento Facial: 20%. Actualización de datos: 10%. Caja de equipos: 10%.								
¿Tiene alguna recomendación sobre el producto final?	Ninguna: 80%. Alertas de acceso a los laboratorios: 10%. Material de la caja de equipos (en acrílico): 10%.								

En la Tabla 3.4 se muestra la cuarta actualización del tablero Kanban, en ella se muestra que las tareas por realizar y las tareas en proceso han finalizado.

Tabla 3.4. Cuarta Actualización Tablero Kanban

TABLERO KANBAN		
TAREAS POR REALIZAR	TAREAS EN PROCESO	TAREAS FINALIZADAS
		Análisis situación actual laboratorios de la FIEE.
		Selección metodología de Desarrollo Software.
		Encuesta de requerimientos.
		Diagramas de casos de uso.
		Diagramas de secuencia.
		Diagramas de actividades.
		Diagramas de clase.
		Diagramas entidad relación.
		Requerimientos físicos del sistema de seguridad.
		Instalación Visual Studio 2017.
		Instalación Librerías Ozeki, EmguCV, OpenCV, ZKTeco e iText.
		Creación de los formularios de la interfaz gráfica.
		Implementación de cada módulo.
		Configuración de los terminales de acceso.
		Configuraciones de equipos intermedios.
		Ubicación física de los terminales de acceso.
		Extensión del cableado necesario para la comunicación entre los equipos.
		Comprobaciones de funcionamiento de los módulos
		Correcciones de los módulos.
		Verificar la existencia de errores producidos al interconectar los equipos.
		Depurar los errores producidos al interconectar los equipos.
		Probar el funcionamiento de las cámaras, lectores de huellas y pines de acceso numérico.
		Se realizarán las encuestas de satisfacción.

4. CONCLUSIONES Y RECOMENDACIONES

Las personas siempre han estado en la búsqueda de aumentar la seguridad de todos los bienes que posee, creando nuevas tecnologías o mejorando las tecnologías existentes para precautelar de mejor forma sus recursos. Este proyecto hace uso de las tecnologías biométricas para mejorar el sistema de seguridad existente en los laboratorios de Redes de la FIEE de la EPN; tras el diseño, desarrollo e implementación del sistema se ha logrado obtener varias conclusiones y recomendaciones que se muestran en los siguientes apartados.

4.1. CONCLUSIONES

- Este sistema ayuda a mejorar la seguridad ya existente en los laboratorios de redes de la FIEE de la EPN; ya que implementa un registro de usuario y una cerradura magnética a las seguridades físicas ya existentes.
- El uso de librerías de código abierto como OpenCV y EmguCV es muy útil ya que no solo permiten la implementación de sistemas basados en la fisiología de las personas; sino que también permiten la implementación de detección de movimiento, reconocimiento de objetos, entre otros.
- Se pudo comprobar que Visual Studio puede trabajar con otros lenguajes como lo es el lenguaje C de Arduino que se utilizó para manipular las cerraduras magnéticas.
- De los tres métodos implementados en el sistema de seguridad se comprobó que el PIN de acceso es el que tiene mayor rendimiento (con un 100%), seguido del reconocimiento dactilar (con un 88%) y finalmente, el reconocimiento Facial con un 68%.
- Si bien el PIN de acceso es el que posee mayor rendimiento, es el menos seguro ya que este puede ser obtenido por otras personas, dándole así acceso al sistema a usuarios desconocidos; por lo cual se debería acompañar al PIN de acceso con el uso de una segunda identificación como la huella dactilar o el reconocimiento facial.
- La fiabilidad de las huellas dactilares está entre las más altas, los errores producidos con este método son principalmente por no tomar una adecuada muestra de la huella durante el registro del usuario o durante el enrolamiento del usuario (muestra de huella para el acceso).

- El reconocimiento facial tiene un rendimiento bajo, en comparación con los otros métodos, debido principalmente a la iluminación donde se encuentran los equipos de acceso, lo que impide la detección facial y por ende el reconocimiento facial.
- Existen problemas al migrar la aplicación de escritorio del computador donde fue desarrollada a un nuevo computador, lo cual dificulta la creación de un instalador de la aplicación de escritorio. Para poder migrar la aplicación de escritorio a un nuevo ordenador es necesario de tener conocimientos de base de datos y de programación para poder modificar la aplicación al nuevo ordenador.
- Los sistemas biométricos, son una de las tecnologías más usadas a nivel mundial, estas proporcionan mayor seguridad a los usuarios que las seguridades típicas como contraseñas, patrones de desbloqueo, pin de acceso, etc.; ya que las características físicas de una persona son difíciles de clonar.
- Durante el estudio de las técnicas para detección y reconocimiento facial y dactilar, se pudo entender de la gran complejidad que poseen los algoritmos biométricos; estos algoritmos están compuestos por altos conceptos y fórmulas matemáticas que combinadas con matrices ayudan a mejorar la seguridad de las personas.
- La metodología de desarrollo Kanban es muy útil al separar las tareas en grupos; ya que esto ayuda a distribuir el trabajo de una manera más eficiente y a reconocer las tareas que ya finalizaron y las que están en proceso.

4.2. RECOMENDACIONES

- Para un correcto funcionamiento del reconocimiento facial es recomendable instalar una adecuada iluminación que se encuentre por encima de los 300 luxes.
- Es recomendable tener actualizados los frameworks de Windows, SQL Server y Visual Studio para evitar los problemas migración de la aplicación de un computador a otro.
- Se debe tener cuidado con los dispositivos de acceso ya que éstos son propensos a daños y al mal uso de los usuarios. Por ejemplo: es recomendable tener limpia la pantalla del lector dactilar para evitar lecturas incorrectas de la huella digital.
- Es muy importante tener actualizada la base de datos; ya que las personas son vulnerables a cambios o daños físicos que pueden de alguna manera cambiar la estructura física de la persona e impedir el uso del sistema de seguridad.
- Se debe establecer correctamente el direccionamiento de los equipos periféricos, para evitar conflictos en la red con direcciones equivocadas o duplicadas.

- Es recomendable completar este proyecto con un estudio de vialidad económica, con respecto a otros sistemas de seguridad que ya se encuentran desarrollados.
- Es recomendable tener actualizadas las librerías que se usaron para el desarrollo de este proyecto para que la aplicación no tenga problemas al ejecutarse en el futuro.
- Se debe tener cuidado con la manipulación de la base de datos; ya que al registrar o actualizar usuarios puede haber conflicto con las claves primarias.
- Es recomendable tener la base de datos en un servidor externo; ya que esto aumenta la seguridad del sistema y lo hace menos vulnerable a ataques.

5. REFERENCIAS BIBLIOGRÁFICAS

- [1] M. Jiménez, J. Bolaños, J. Serrano, "Diseño e implementación de un sistema de seguridad basado en reconocimiento de rostros", Escuela Politécnica Nacional, Quito – Ecuador, 2005. [Online]. Disponible en: <https://bibdigital.epn.edu.ec/handle/15000/5504>
- [2] O. Cerón, H. Coronel, "Reconocimiento de Rostros Utilizando Redes Neuronales", Escuela Politécnica Nacional, Quito – Ecuador, 2007. [Online]. Disponible en: <https://bibdigital.epn.edu.ec/handle/15000/5506>
- [3] O. Mayorga, R. Carrera, "Estudio comparativo de LINQ y expresiones lambda como paradigmas de programación en el sistema “HADE” aplicado a COMPROTEC-ESPOCH", Escuela Superior Politécnica de Chimborazo, Riobamba – Ecuador, 2010. [Online]. Disponible en: <http://dspace.esepoch.edu.ec/handle/123456789/3633?mode=full>
- [4] D. Espinoza, P. Jorquera, "Reconocimiento Facial", Pontificia Universidad Católica de Valparaíso, Valparaíso – Chile, 2015. [Online]. Disponible en: http://opac.pucv.cl/pucv_txt/txt-1000/UCD1453_01.pdf
- [5] J. Pilataxi, L. Morales, H. Espinosa, "Diseño e implementación de un sistema de seguridad y alerta para vehículos, basado en reconocimiento facial y localización GPS, en una Raspberry Pi B plus", Escuela Politécnica Nacional, Quito – Ecuador, 2016. [Online]. Disponible en: <https://bibdigital.epn.edu.ec/handle/15000/16667>
- [6] A. Pătrașcu, "Aplicación para Detección y Reconocimiento Facial en Interiores", Escuela Técnica Superior de Ingeniería, Universidad de Sevilla, Sevilla – España, 2016. [Online]. Disponible en: http://bibing.us.es/proyectos/abreprojy/90722/fichero/ResumenTFG_PatrascuVioricaAndreea.pdf
- [7] V. Mosquera, E. Romero, "Diseño de un software piloto de reconocimiento facial para el control de asistencia en la escuela de telecomunicaciones de la Universidad de Carabobo", Universidad de Carabobo, Carabobo – Venezuela, 2016. [Online]. Disponible en: <http://mriuc.bc.uc.edu.ve/bitstream/handle/123456789/5232/vmosquera.pdf>
- [8] S. Domínguez, "Reconocimiento facial mediante el Análisis de Componentes Principales (PCA)", Escuela Técnica Superior de Ingeniería, Universidad de Sevilla, Sevilla – España, 2017. [Online]. Disponible en: <https://idus.us.es/xmlui/handle/11441/66514>

- [9] J. Krajewski, P. Ritzman, "Administración De Operaciones: Estrategia y Análisis", 5ta edición, Pearson Education Company, México 2000. Páginas 743 – 745. [Online]. Disponible en: http://biblio.econ.uba.ar/opac-tmpl/bootstrap/tc/145279_TC.pdf
- [10] D. Grundgeiger, "Programming Visual Basic .NET", 1st edición. O'Reilly Publisher. Enero 2002. Páginas 13 – 16. [Online]. Disponible en: https://www.visualchart.com/ContentManagement/Development/Manuals/EN/vbNet_programming.pdf
- [11] J. Vélez, A. Moreno, Á. Sánchez, J. Sánchez – Marín, "Visión Por Computadora", 1ra edición, Dykinso S.L, 2003. Páginas 28 – 32. [Online]. Disponible: <http://www.visionporcomputador.es/libroVision/libro.html>
- [12] R. Capms, L. Casillas, D. Costal, M. Gibert, C. Martín, O. Pérez, "Base de Datos", 1st edición. Eureka Media, SL. Mayo 2005. Páginas 113 – 118. [Online]. Disponible en: http://www.sw-computacion.f2s.com/Linux/007-Bases_de_datos.pdf
- [13] G. Bradski, A. Kaehler, "Learning OpenCV". 1st edición. O'Reilly Publisher. Septiembre 2008. Páginas 1 – 8. [Online]. Disponible en: <https://www.bogotobogo.com/cplusplus/files/OReilly%20Learning%20OpenCV.pdf>
- [14] A. Oppel, R. Sheldon. "Fundamentos de SQL". 3ra edición. Mc Graw Hill Educación. México 2010. Páginas 5 – 18. Traducción: C. Jiménez. [Online]. Disponible en: <https://www.elsolucionario.org/fundamentos-de-sql-oppel-r-sheldon-3ed/>
- [15] Federico G. Rudolph, "Introducción a Visual Studio. NET". 1st edición. Manual de Referencia, Curso de Capacitación en .NET. 2010. Páginas 1 – 6. [Online]. Disponible en: <https://es.calameo.com/read/005274837376a635a826b>
- [16] C. Tolosa, A. Giz, "Sistemas Biométricos". Anónimo. Anónimo. Páginas 17 – 19 y Páginas 22 – 23. [Online]. Disponible en: https://www.dsi.uclm.es/personal/MiguelFGraciani/mikicurri/Docencia/Bioinformatica/web_BIO/Documentacion/Trabajos/Biometria/Trabajo%20Biometria.pdf
- [17] P. Viola, M. Jones, "Rapid Object Detection using a Boosted Cascade as Simple Features". Computer vision and pattern recognition. 2001. Páginas 1 – 9. [Online]. Disponible en: <https://www.cs.cmu.edu/~efros/courses/LBMV07/Papers/viola-cvpr-01.pdf>
- [18] A. Rosales, "Clasificación de Huellas Digitales Mediante Minucias". Instituto Nacional de Astrofísica, Óptica y Electrónica, abril 2009. Páginas 1 – 9. [Online]. Disponible en: https://ccc.inaoep.mx/~esucar/Clases-mgp/Proyectos/reportes_modelos_huellas.pdf

- [19] M. Ruiz, J. Rodríguez, J. Olivares, "A glance to the biometric". Revista Avances en Sistemas Informáticos" Vol. 06, No 02, febrero 2009. Páginas 1 – 10. [Online]. Disponible en: <http://www.bdigital.unal.edu.co/23395/1/20295-68748-1-PB.pdf>
- [20] J. Mena, "Acceso a datos con LINQ to SQL". Revista Telem@tica. Vol. 11. No. 2, mayo-agosto, 2012, Páginas 1 – 9. [Online]. Disponible en: <https://pdfs.semanticscholar.org/994f/8141cb859437bf418f1a70ba78b58cee94af.pdf>
- [21] C. Rivas, V. Corona, J. Gutiérrez y L. Hernández, "Metodologías Actuales De Desarrollo Software". Revista Tecnología e Investigación. Vol. 002. No. 5, diciembre 2015, Páginas 980 – 986. [Online]. Disponible en: http://www.ecorfan.org/bolivia/researchjournals/Tecnologia_e_innovacion/vol2num5/Tecnologia_e_Innovacion_Vol2_Num5_6.pdf
- [22] Anónimo, "Tecnologías biométricas aplicadas a la ciberseguridad". Revista del Instituto Nacional de Ciberseguridad. Vol. 01. No. 06. 2016, Páginas 4 – 30. [Online]. Disponible en: https://www.incibe.es/sites/default/files/contenidos/guias/doc/guia_tecnologias_biometricas_aplicadas_ciberseguridad_metad.pdf
- [23] M. Paredes, M. Valle, G. Alvarón, F. Huincho, K. Gutiérrez, "Sistema de vigilancia biométrico para el control delincuencial en la división policial". Revista Conocimiento para el Desarrollo. 2017, Paginas 1 – 7. [Online]. Disponible en: <http://repositorio.usanpedro.edu.pe/bitstream/handle/USANPEDRO/293/PI1610091.PDF?sequence=1&isAllowed=y>
- [24] Universidad Continental. (2019, septiembre 16). "Estudiantes de Universidad Continental crean software para detectar rostros y prevenir inseguridad ciudadana" [Online]. Disponible en: <https://orientacion.universia.edu.pe/universidades/universidad-continental-102/noticias/estudiantes-de-universidad-continental-crean-software-para-detectar-rostros-y-prevenir-inseguridad-ciudadana-2550.html#>
- [25] Cognitive Services. (2019, septiembre 16). "Face++ 101: How Do I Find the One with Face Search?" [Online]. Disponible en: <https://www.faceplusplus.com/blog/article/face-101-how-do-i-find-the-one/>
- [26] SlideShare. (2019, septiembre 16). "Diseñando Sistemas empleando el modelo de capas en desarrollo de software". Por: Ernesto Alexander Calderón Peraza. Docente del Departamento de Ingeniería. Área de Informática [Online]. Disponible en: <https://es.slideshare.net/calderonperaza/disenando-sistemas-empleando-el-modelo-de-capas-en-desarrollo-de-software>

- [27] FujiFilm. (2019, septiembre 16). “Cámaras Digitales - Detección facial integrada en cámaras FujiFilm” [Online]. Disponible en: https://es.fujifilmusa.com/products/digital_cameras/a/finepix_ax660/features/img/page_03/pic_07.jpg
- [28] Wikipedia. (2019, septiembre 16). “Microsoft Visual Studio” [Online]. Disponible en: https://es.wikipedia.org/wiki/Microsoft_Visual_Studio
- [29] Visual Studio. (2019, julio 2019). “Aprendizaje Visual Studio 2013” [Online]. Disponible en: <https://riptutorial.com/Download/visual-studio-2013-es.pdf>
- [30] Microsoft Docs. (2019, septiembre 16). “Novedades de Visual Studio 2017” [Online]. Disponible en: <https://docs.microsoft.com/es-es/visualstudio/ide/whats-new-visual-studio-2017?view=vs-2017>
- [31] Microsoft Docs. (2019, septiembre 16). “Requisitos del sistema de la familia de productos Visual Studio 2017” [Online]. Disponible en: <https://docs.microsoft.com/en-us/visualstudio/productinfo/vs2017-system-requirements-vs#visual-studio-2017-system-requirements>
- [32] Wikipedia. (2019, septiembre 16). “Microsoft .NET” [Online]. Disponible en: https://es.wikipedia.org/wiki/Microsoft_.NET
- [33] Microsoft Docs. (2019, septiembre 16). “Compatibilidad y destinatarios de la plataforma Visual Studio 2017” [Online]. Disponible: <https://docs.microsoft.com/es-es/visualstudio/productinfo/vs2017-compatibility-vs>
- [34] Microsoft Docs. (2019, septiembre 16). “Conceptos básicos de Windows Communication Foundation” [Online]. Disponible en: <https://docs.microsoft.com/es-es/dotnet/framework/wcf/fundamental-concepts>
- [35] Microsoft Docs. (2019, septiembre 16). “¿Qué es Windows Communication Foundation?” [Online]. Disponible en: <https://docs.microsoft.com/es-es/dotnet/framework/wcf/whats-wcf>
- [36] WCF Tutorial.net (2019, juli19). “Introduction to WCF” [Online]. Disponible en: <http://www.wcftutorial.net/Introduction-to-WCF.aspx>
- [37] Microsoft Docs. (2019, septiembre 16). “Introducción al lenguaje C# y .NET Framework” [Online]. Disponible en: <https://docs.microsoft.com/es-es/dotnet/csharp/getting-started/introduction-to-the-csharp-language-and-the-net-framework>
- [38] Microsoft Docs. (2019, septiembre 16). “Resumen de tipos de datos (Visual Basic)” [Online]. Disponible en: <https://docs.microsoft.com/es-es/dotnet/visual-basic/language-reference/data-types/>

- [39] CodeProject. (2019, septiembre 16). “Linq DLinq XLinq PLinq All at one place” [Online]. Disponible en: <https://www.codeproject.com/Articles/105098/Linq-DLinq-XLinq-PLinq-All-at-one-place>
- [40] Programar Fácil. (2019, septiembre 16). “¿Qué es un ORM?” [Online]. Disponible en: <https://programarfacil.com/blog/que-es-un-orm/>
- [41] OpenCV. (2019, septiembre 16). “OpenCV” [Online]. Disponible en: <https://opencv.org/>
- [42] GNU Operating System. (2019, septiembre 16). “El problema de la licencia BSD” [Online]. Disponible en: <https://www.gnu.org/licenses/bsd.html>
- [43] EmguCV. (2019, septiembre 16). “EmguCV” [Online]. Disponible en: http://www.emgu.com/wiki/index.php/Main_Page
- [44] Ozeki Camera SDK. (2019, septiembre 16). “Ozeki Camera SDK - Product Guide” [Online]. Disponible en: http://www.camera-sdk.com/p_12-quick-start-guide-for-the-ozeki-camera-sdk-onvif.html
- [45] ONVIF. (2019, septiembre 16). “Onvif Organization” [Online]. Disponible en: <https://www.onvif.org/about/organization/>
- [46] Ozeki Camera SDK. (2019, septiembre 16). “Ozeki Camera SDK – Product Guide” [Online]. Disponible en: http://www.camera-sdk.com/p_12-quick-start-guide-for-the-ozeki-camera-sdk-onvif.html
- [47] Equifer. (2019, septiembre 16). “Lectores Biométricos” [Online]. Disponible en: <http://www.equifer.com/lectores-biometricos/>
- [48] SAD UT3. (2019, juli19). “Biometría” [Online]. Disponible en: http://dis.um.es/~lopezquesada/documentos/IES_1213/SAD/curso/UT3/ActividadesAlumnos/2/html/biometria.html
- [49] Medium. (2019, septiembre 16). “Reconocimiento facial: Entiendo el algoritmo LBPH” [Online]. Disponible en: <https://towardsdatascience.com/face-recognition-how-lbph-works-90ec258c3d6b>
- [50] PublicaTIC. (2019, julio, 2019). “Kanban as a process for agile development” [Online]. Disponible en: <https://blogs.deusto.es/master-informatica/kanban-as-a-process-for-agile-development/>
- [51] AHMED. (2019, septiembre 16). “Face Detection and Recognition in C# using EmguCV 3.0 (OpenCV Wrapper)” [Online]. Disponible en: <http://ahmedopeyemi.com/main/face-detection-and-recognition-in-c-using-emgucv-3-0-opencv-wrapper-part-1/>
- [52] CodeProject. (2019, septiembre 16). “Camera Face Detection in C# using Emgu CV and WPF” [Online]. Disponible en:

- http://www.codeproject.com/Articles/462527/Camera-Face-Detection-in-Csharp-Using-Emgu-CV-Open
- [53] CodeProject. (2019, septiembre 16). “Multiple Face Detection and Recognition in Real Time” [Online]. Disponible en: <http://www.codeproject.com/Articles/239849/Multiple-face-detection-and-recognition-in-real>
- [54] Ozeki Camera SDK. (2019, septiembre 16). “How to Implement Face Detection in C#” [Online]. Disponible en: http://www.camera-sdk.com/p_267-how-to-implement-face-detection-in-c-onvif.html
- [55] Emgu. (2019, septiembre 16). “Face Detection” [Online]. Disponible en: http://www.emgu.com/wiki/index.php/Face_detection
- [56] CodeProject. (2019, septiembre 16). “Using Raw Input from C# to handle multiple keyboards” [Online]. Disponible en: <https://www.codeproject.com/Articles/17123/Using-Raw-Input-from-C-to-handle-multiple-keyboard>
- [57] ZKTeco. (2019, septiembre 16). “OnEnroll” [Online]. Disponible en: http://desarrollo.zktecolatinoamerica.com/sdk/on_enroll

ANEXOS

- ANEXO A. Requisitos del sistema de Visual Studio 2017.
- ANEXO B. Video “OpenCV Face Detection: Visualized.” para la Detección Facial.
- ANEXO C. Encuesta de los Requerimientos Funcionales y No Funcionales.
- ANEXO D. Diagrama de clases de la interfaz de usuario.
- ANEXO E. Manual de usuario.
- ANEXO F. Encuesta de Satisfacción sobre el sistema prototipo.

ANEXO A

Sistemas operativos compatibles	<p>Visual Studio 2017 se instalará y ejecutará en los siguientes sistemas operativos:</p> <ul style="list-style-type: none">• Windows 10 versión 1507 o superior: Hogar, Profesional, Educación y Empresa (no se admiten LTSC y S)• Windows Server 2016: estándar y centro de datos• Windows 8.1 (con la actualización 2919355): Core, Professional y Enterprise• Windows Server 2012 R2 (con la actualización 2919355): Essentials, Standard, Datacenter• Windows 7 SP1 (con las últimas actualizaciones de Windows): Home Premium, Professional, Enterprise, Ultimate
Hardware	<ul style="list-style-type: none">• Procesador de 1.8 GHz o más rápido. Dual-core o mejor recomendado• 2 GB de RAM; Se recomiendan 4 GB de RAM (2,5 GB como mínimo si se ejecuta en una máquina virtual)• Espacio en disco duro: hasta 130 GB de espacio disponible, según las características instaladas; Las instalaciones típicas requieren 20-50 GB de espacio libre.• Velocidad del disco duro: para mejorar el rendimiento, instale Windows y Visual Studio en una unidad de estado sólido (SSD).• Tarjeta de video que admite una resolución de pantalla mínima de 720p (1280 por 720); Visual Studio funcionará mejor a una resolución de WXGA (1366 por 768) o superior.
Idiomas soportados	<p>Visual Studio está disponible en inglés, chino (simplificado), chino (tradicional), checo, francés, alemán, italiano, japonés, coreano, polaco, portugués (Brasil), ruso, español y turco.</p> <p>Puede seleccionar el idioma de Visual Studio durante la instalación. El Visual Studio Installer está disponible en los mismos catorce idiomas y coincidirá con el idioma de Windows, si está disponible.</p> <p>Nota: Visual Studio Team Foundation Server Office Integration 2017 está disponible en los diez idiomas compatibles con Visual Studio Team Foundation Server 2017.</p>
Requerimientos adicionales	<ul style="list-style-type: none">• Se requieren derechos de administrador para instalar Visual Studio.• Se requiere .NET Framework 4.5 para instalar Visual Studio. Visual Studio requiere .NET Framework 4.6.1, que se instalará durante la instalación.• Windows 10 Enterprise LTSC edition y Windows 10 S no son compatibles con el desarrollo. Puede usar Visual Studio 2017 para crear aplicaciones que se ejecuten en Windows 10 LTSC y Windows 10 S.• Se requiere Internet Explorer 11 o Edge para escenarios relacionados con Internet. Es posible que algunas funciones no funcionen a menos que se instalen estas o una versión posterior.• Para la compatibilidad con emuladores, se requieren las ediciones Windows 8.1 Pro o Enterprise (x64). También se requiere un procesador que admita Client Hyper-V y la traducción de direcciones de segundo nivel (SLAT).• El desarrollo de la aplicación universal de Windows, incluido el diseño, la edición y la depuración, requiere Windows 10. Windows Server 2016 y Windows Server 2012 R2 se pueden usar para crear aplicaciones de Universal Windows desde la línea de comandos.• Las opciones de Server Core y Minimal Server Interface no son compatibles cuando se ejecuta Windows Server.• Los contenedores de Windows no son compatibles, excepto para Visual Studio 2017 Build Tools.• Team Foundation Server 2017 Office Integration requiere Office 2016, Office 2013 u Office 2010.• Xamarin.Android requiere una edición de 64 bits de Windows y el Kit de desarrollo de Java de 64 bits (JDK).• Se requiere PowerShell 3.0 o superior en Windows 7 SP1 para instalar Mobile Development con cargas de trabajo de C++, JavaScript o .NET.

ANEXO B

Se proporciona el enlace al video: <https://vimeo.com/12774628>

ANEXO C

El formato de la encuesta de requerimientos y las encuestas realizadas se encuentran adjuntas en el CD.

ANEXO D

La imagen del diagrama de clases de la interfaz de usuario se encuentra adjunta en el CD.

ANEXO E

El manual de usuario se encuentra adjunto en el CD.

ANEXO F

El formato de la encuesta de satisfacción y las encuestas realizadas se encuentran adjuntas en el CD.

ORDEN DE EMPASTADO