

	GESTIÓN SERVICIOS BIBLIOTECARIOS						  
	CARTA DE AUTORIZACIÓN						
CÓDIGO	AP-BIB-FO-06	VERSIÓN	1	VIGENCIA	2014	PÁGINA	1 de 2

Neiva, 21 de junio de 2016

Señores

CENTRO DE INFORMACIÓN Y DOCUMENTACIÓN

UNIVERSIDAD SURCOLOMBIANA

Ciudad

El (Los) suscrito(s):

Yeison Gerardo Ortiz Palomá, con C.C. No. 1075231914,
Andres Felipe Cordoba Villegas, con C.C. No. 1075239890,
_____, con C.C. No. _____,
_____, con C.C. No. _____,

autor(es) de la tesis y/o trabajo de grado o _____

titulado Diseño de prototipo de un sistema de gestión para un gimnasio utilizando tecnología NFC

presentado y aprobado en el año 2016 como requisito para optar al título de

Ingeniero Electrónico _____;

autorizo (amos) al CENTRO DE INFORMACIÓN Y DOCUMENTACIÓN de la Universidad Surcolombiana para que con fines académicos, muestre al país y el exterior la producción intelectual de la Universidad Surcolombiana, a través de la visibilidad de su contenido de la siguiente manera:

Los usuarios puedan consultar el contenido de este trabajo de grado en los sitios web que administra la Universidad, en bases de datos, repositorio digital, catálogos y en otros sitios web, redes y sistemas de información nacionales e internacionales "open access" y en las redes de información con las cuales tenga convenio la Institución.

- Permita la consulta, la reproducción y préstamo a los usuarios interesados en el contenido de este trabajo, para todos los usos que tengan finalidad académica, ya sea en formato Cd-Rom o digital desde internet, intranet, etc., y en general para cualquier formato conocido o por conocer, dentro de los términos establecidos en la Ley 23 de 1982, Ley 44 de 1993, Decisión Andina 351 de 1993, Decreto 460 de 1995 y demás normas generales sobre la materia.

	GESTIÓN SERVICIOS BIBLIOTECARIOS					  	
	CARTA DE AUTORIZACIÓN						
CÓDIGO	AP-BIB-FO-06	VERSIÓN	1	VIGENCIA	2014	PÁGINA	2 de 2

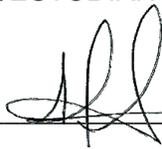
• Continúo conservando los correspondientes derechos sin modificación o restricción alguna; puesto que de acuerdo con la legislación colombiana aplicable, el presente es un acuerdo jurídico que en ningún caso conlleva la enajenación del derecho de autor y sus conexos.

De conformidad con lo establecido en el artículo 30 de la Ley 23 de 1982 y el artículo 11 de la Decisión Andina 351 de 1993, "Los derechos morales sobre el trabajo son propiedad de los autores", los cuales son irrenunciables, imprescriptibles, inembargables e inalienables.

EL AUTOR/ESTUDIANTE:

Firma: Yeison Ortiz

EL AUTOR/ESTUDIANTE:

Firma: 

EL AUTOR/ESTUDIANTE:

Firma: _____

EL AUTOR/ESTUDIANTE:

Firma: _____

	GESTIÓN SERVICIOS BIBLIOTECARIOS						  
	DESCRIPCIÓN DE LA TESIS Y/O TRABAJOS DE GRADO						
CÓDIGO	AP-BIB-FO-07	VERSIÓN	1	VIGENCIA	2014	PÁGINA	1 de 3

TÍTULO COMPLETO DEL TRABAJO: Diseño de prototipo de un sistema de gestión para un gimnasio utilizando tecnología NFC

AUTOR O AUTORES:

Primero y Segundo Apellido	Primero y Segundo Nombre
Ortiz Palomá	Yeison Gerardo
Cordoba Villegas	Andres Felipe

DIRECTOR Y CODIRECTOR TESIS:

Primero y Segundo Apellido	Primero y Segundo Nombre
Bravo Obando	Martin Diomedes

ASESOR (ES):

Primero y Segundo Apellido	Primero y Segundo Nombre
Molina Mosquera	Johan Julian
Quintero Polanco	Jesus David

PARA OPTAR AL TÍTULO DE: Ingeniero Electrónico

FACULTAD: Ingeniería

PROGRAMA O POSGRADO: Electrónica

CIUDAD: Neiva

AÑO DE PRESENTACIÓN: 2016

NÚMERO DE PÁGINAS: 136

TIPO DE ILUSTRACIONES (Marcar con una X):

Diagramas X Fotografías X Grabaciones en discos ___ Ilustraciones en general ___ Grabados ___ Láminas ___
Litografías ___ Mapas ___ Música impresa ___ Planos X Retratos ___ Sin ilustraciones ___ Tablas o Cuadros X

SOFTWARE requerido y/o especializado para la lectura del documento: PDF (formato de documento portátil)

	GESTIÓN SERVICIOS BIBLIOTECARIOS					  	
	DESCRIPCIÓN DE LA TESIS Y/O TRABAJOS DE GRADO						
CÓDIGO	AP-BIB-FO-07	VERSIÓN	1	VIGENCIA	2014	PÁGINA	2 de 3

MATERIAL ANEXO: Instaladores GymID

PREMIO O DISTINCIÓN (En caso de ser LAUREADAS o Meritoria):

PALABRAS CLAVES EN ESPAÑOL E INGLÉS:

<u>Español</u>	<u>Inglés</u>	<u>Español</u>	<u>Inglés</u>
1. <u>NFC</u>	<u>NFC</u>	6. <u>Banco de datos</u>	<u>Database</u>
2. <u>Radio frecuencia</u>	<u>Radiofrequency</u>	7. <u>Software libre</u>	<u>Free software</u>
3. <u>Sistema de identificación</u>	<u>Identification system</u>	8. _____	_____
4. <u>Administración de Gimnasio</u>	<u>Gym administration</u>	9. _____	_____
5. <u>UML</u>	<u>UML</u>	10. _____	_____

RESUMEN DEL CONTENIDO: (Máximo 250 palabras)

La implementación de nuevas tecnologías ha acelerado a grandes pasos el desarrollo tecnológico mundial, NFC no es una tecnología relativamente nueva, sin embargo, con la implementación en teléfonos móviles y la necesidad de fortalecer la seguridad y velocidad de los sistemas de identificación, la tecnología NFC ha vuelto a tener protagonismo. Este artículo plantea el diseño e implementación de un sistema de gestión para un gimnasio utilizando la tecnología NFC que permite optimizar el redimiendo de los servicios ofrecidos por un gimnasio a un bajo costo, dando la posibilidad que este tipo de negocios mejoren la organización de sus diferentes procesos. La implementación del sistema ofrece una novedosa forma de gestionar la entrada de clientes al gimnasio, permitiendo al cliente identificarse por medio de tarjetas NFC que mejorará significativamente la velocidad de acceso al gimnasio. Adicionalmente el sistema incluye una aplicación para escritorio que cuenta con una completa lista de herramientas que administran toda la información de clientes, pagos, membresías, ejercicios y rutinas de entrenamiento.

ABSTRACT: (Máximo 250 palabras)



GESTIÓN SERVICIOS BIBLIOTECARIOS

DESCRIPCIÓN DE LA TESIS Y/O TRABAJOS DE GRADO



CÓDIGO

AP-BIB-FO-07

VERSIÓN

1

VIGENCIA

2014

PÁGINA

3 de 3

The implementation of new technologies has accelerated strides global technological development, NFC is not a relatively new technology, however, with the implementation in mobile phones and the need to strengthen security and speed identification systems, NFC has returned to be important. This paper presents the design and implementation of a management system for a gym using the NFC technology that optimizes the redeeming of the services offered by a gym at a low cost, giving the possibility that these businesses improve the organization of their different processes. The implemented system offers a new way to manage customer entrance to the gym, allowing the customer identified through NFC cards will significantly improve the speed of access to the gym. Additionally, the system includes a desktop application that has a complete list of tools that manage all customer information, payments, memberships, exercises and training routine.

APROBACION DE LA TESIS

Nombre Presidente Jurado: Martin Diomedes Bravo Obando

Firma:

Nombre Jurado: Johan Julian Molina Mosquera

Firma:

Nombre Jurado: Jesus David Quintero Polanco

Firma:

DISEÑO DE PROTOTIPO DE UN SISTEMA DE GESTIÓN PARA UN GIMNASIO UTILIZANDO TECNOLOGÍA NFC

YEISON GERARDO ORTIZ PALOMÁ
ANDRES FELIPE CORDOBA VILLEGAS



UNIVERSIDAD SURCOLOMBIANA
FACULTAD DE INGENIERÍA
INGENIERÍA ELECTRÓNICA
NEIVA
2016

DISEÑO DE PROTOTIPO DE UN SISTEMA DE GESTIÓN PARA UN GIMNASIO UTILIZANDO TECNOLOGÍA NFC

YEISON GERARDO ORTIZ PALOMÁ
ANDRES FELIPE CORDOBA VILLEGAS

Proyecto de grado para optar al título de Ingeniero Electrónico

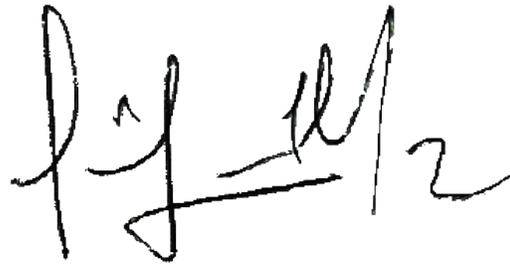
Director
Martín Diomedes Bravo Obando
Ingeniero Electrónico



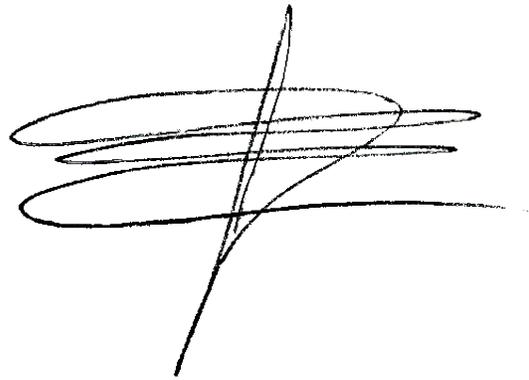
UNIVERSIDAD SURCOLOMBIANA
FACULTAD DE INGENIERÍA
INGENIERÍA ELECTRÓNICA
NEIVA
2016

Nota de aceptación:

Aprobado por el Comité de Grado en cumplimiento de los requisitos exigidos por la Universidad Surcolombiana para optar al título de Ingeniero Electrónico



Jurado: Johan Julian Molina Mosquera



Jurado: Jesus David Quintero Polanco

Neiva, 7 de Marzo de 2016

AGRADECIMIENTOS

A nuestros padres, familiares y amigos quienes estuvieron siempre apoyando incondicionalmente nuestro trabajo y dándonos fuerzas para seguir adelante, especialmente Martha Villegas, Mario Cordoba, Julia Palomá, Adelaida Palomá y Luis Rojas.

Igualmente agradecemos a todos los maestros que hicieron parte de nuestro proceso de formación, a nuestro director Martín Bravo por asesorarnos durante todo el transcurso del proyecto, a la firma Everkinetic por suministrarnos las imágenes para crear la base de datos de ejercicios para la aplicación GymID.

CONTENIDO

GLOSARIO	15
RESUMEN	16
INTRODUCCIÓN	17
1. ANÁLISIS DE REQUERIMIENTOS	19
1.1. PLANTEAMIENTO DEL PROBLEMA	19
1.2. PLANTEAMIENTO DEL CASO	19
1.3. ACTORES	20
1.4. DIAGRAMAS DE CASOS DE USO	22
1.4.1. Diagrama de casos de uso del sistema.	22
1.4.2. Diagrama de casos de uso para el Administrador.	22
1.4.3. Diagrama de casos de uso para el Cliente.	23
1.4.4. Diagrama de casos de uso para el Instructor.	24
1.4.5. Diagrama de casos de uso para el Recepcionista.	24
1.5. ALCANCES	25
1.6. LIMITACIONES	26
1.7. ARQUITECTURA DEL SISTEMA	26
1.7.1. Arquitectura del hardware.	27
1.7.2. Arquitectura del software.	30

2. DISEÑO	32
2.1. CLASES DEL SISTEMA	32
2.1.1. Diagrama de herencia en actores.	32
2.1.2. Diagrama de clases.	33
2.1.3. Clase Recepcionista.	34
2.1.4. Clase Pago.	35
2.1.5. Clase PC.	35
2.1.6. Clase GUI.	36
2.1.7. Clase Base de Datos.	36
2.1.8. Clase Arduino.	37
2.1.9. Clase Display LCD.	37
2.1.10. Clase Lector NFC.	38
2.1.11. Clase Identificación.	38
2.1.12. Clase Cliente.	38
2.1.13. Clase Instructor.	39
2.1.14. Clase Rutina de Entrenamiento.	39
2.1.15. Clase Ejercicio.	40
2.1.16. Clase Administrador.	40
2.1.17. Clase Membresía.	41
2.2. DIAGRAMAS DE SECUENCIAS	41
2.2.1. Administrar Membresía.	41
2.2.2. Administrar Usuario.	43
2.2.3. Generar PDF de historial de registros.	45
2.2.4. Identificar.	46

CONTENIDO

2.2.5.	Conocer estado de cuenta.	47
2.2.6.	Administrar Ejercicio.	47
2.2.7.	Administrar Rutina.	50
2.2.8.	Administrar Cliente.	51
2.2.9.	Administrar Pago.	54
2.3.	LECTOR DE ETIQUETAS NFC	56
2.3.1.	Arduino UNO R3.	56
2.3.2.	Escudo Adafruit PN532 NFC/RFID.	56
2.3.3.	Pantalla LCD.	56
2.3.4.	LED bicolor.	56
2.3.5.	Escudo de conexión.	56
2.3.6.	Funcionamiento.	57
2.3.7.	Programación.	58
2.3.8.	Escudo de conexión.	60
3.	IMPLEMENTACIÓN	62
3.1.	DIAGRAMA DE DESPLIEGUE	62
3.2.	DIAGRAMA DE COMPONENTES	63
3.3.	BASE DE DATOS	65
3.3.1.	Tablas.	67
3.3.2.	Vistas.	69
3.3.3.	Rutinas.	69
3.3.4.	Disparadores.	69
3.4.	MONTAJE Y PRUEBAS DEL LECTOR DE ETIQUETAS NFC	70

3.5. APLICACIÓN PARA ESCRITORIO	74
3.5.1. Entrar al Sistema.	74
3.5.2. Ventana Principal.	74
3.5.3. Administrar Usuarios.	76
3.5.4. Administrar Membresías.	77
3.5.5. Administrar Clientes	79
3.5.6. Administrar Pagos.	80
3.5.7. Administrar Ejercicios.	82
3.5.8. Administrar Rutinas.	83
3.5.9. Historial de Registros.	84
3.5.10. Perfil del Cliente.	85
3.5.11. Identificar.	88
4. CONCLUSIONES	91
5. RECOMENDACIONES	92
BIBLIOGRAFÍA	93
ANEXOS	95
A. Especificaciones de los Casos de Uso	95
A.1. Casos de uso para el Administrador	96
A.2. Casos de uso para el Cliente	104
A.3. Casos de uso para el Instructor	106
A.4. Casos de uso para el Recepcionista	117

B. Código de Arduino	128
B.1. Declaración de variables.	128
B.2. Preparación setup().	128
B.3. Bucle principal loop().	129
B.4. Función de bienvenida Saludo().	133
C. Diseño de la cubierta del Lector NFC	135

LISTA DE CUADROS

A.1. Diccionario de elementos del sistema	95
A.2. Especificación del caso de uso Administrar Membresía	96
A.3. Especificación del caso de uso Crear Membresía	97
A.4. Especificación del caso de uso Editar Membresía	98
A.5. Especificación del caso de uso Eliminar Membresía	99
A.6. Especificación del caso de uso Administrar Usuario	100
A.7. Especificación del caso de uso Crear Usuario	101
A.8. Especificación del caso de uso Eliminar Usuario	102
A.9. Especificación del caso de uso Generar PDF de historial de registros . .	103
A.10.Especificación del caso de uso Identificar	104
A.11.Especificación del caso de uso Conocer estado de cuenta	105
A.12.Especificación del caso de uso Administrar Ejercicio	106
A.13.Especificación del caso de uso Crear Ejercicio	107
A.14.Especificación del caso de uso Editar Ejercicio	108
A.15.Especificación del caso de uso Eliminar Ejercicio	109
A.16.Especificación del caso de uso Importar Ejercicios	110
A.17.Especificación del caso de uso Exportar Ejercicios	111
A.18.Especificación del caso de uso Administrar Rutina	112
A.19.Especificación del caso de uso Crear Rutina	113

LISTA DE CUADROS

A.20.Especificación del caso de uso Editar Rutina	114
A.21.Especificación del caso de uso Eliminar Rutina	115
A.22.Especificación del caso de uso Generar PDF de rutinas	116
A.23.Especificación del caso de uso Administrar Cliente	117
A.24.Especificación del caso de uso Crear Cliente	118
A.25.Especificación del caso de uso Editar Cliente	119
A.26.Especificación del caso de uso Eliminar Cliente	120
A.27.Especificación del caso de uso Asignar identificaciones	121
A.28.Especificación del caso de uso Tomar Fotografías	122
A.29.Especificación del caso de uso Administrar Pago	123
A.30.Especificación del caso de uso Crear Pago	124
A.31.Especificación del caso de uso Eliminar Pago	125
A.32.Especificación del caso de uso Pagar Saldo	126
A.33.Especificación del caso de uso Identificar	127

LISTA DE FIGURAS

1.1. Diagrama de herencia de los Actores	21
1.2. Diagrama de Casos de uso del Sistema	22
1.3. Diagrama de Casos de uso para el Administrador	23
1.4. Diagrama de Casos de uso para el Cliente	23
1.5. Diagrama de Casos de uso para el Instructor	24
1.6. Diagrama de Casos de uso para el Recepcionista	25
1.7. Etiquetas NFC Mifare	27
1.8. Escudo Adafruit PN532	28
1.9. Comunicación entre PN532 y Etiqueta Mifare	28
1.10. Pantalla LCD	29
1.11. LED Bicolor	30
1.12. Arduino UNO R3	30
2.1. Diagrama de Herencia en Actores del Sistema	33
2.2. Diagrama de Clases del Sistema	34
2.3. Diagrama Secuencias para Administrar Membresía	43
2.4. Diagrama Secuencias para Administrar Usuario	44
2.5. Diagrama Secuencias para Generar PDF Historial de Registros	45
2.6. Diagrama Secuencias para Identificar	46
2.7. Diagrama Secuencias para Conocer Estado de Cuenta	47

LISTA DE FIGURAS

2.8. Diagrama Secuencias para Administrar Ejercicio	49
2.9. Diagrama Secuencias para Administrar Rutina	51
2.10. Diagrama Secuencias para Administrar Cliente	53
2.11. Diagrama Secuencias para Administrar Pago	55
2.12. Estructura física del Lector de etiquetas NFC	56
2.13. Diagrama de Estados del Lector de Etiquetas NFC	57
2.14. Diseño esquemático del Escudo de Conexión	61
2.15. Diseño PCB del Escudo de Conexión	61
3.1. Diagrama de Despliegue del Sistema	62
3.2. Diagrama de Componentes del Sistema	63
3.3. Diagrama de Componentes de GymID.exe	64
3.4. Modelo relacional del banco de datos	66
3.5. Escudo de conexión	71
3.6. Montaje del lector de etiquetas NFC	71
3.7. Lector de etiquetas NFC en funcionamiento	72
3.8. Formato de posibles lecturas	73
3.9. Interfaz de Entrar al Sistema	74
3.10. Interfaz de la Ventana Principal	75
3.11. Permisos del Usuario Administrador	76
3.12. Permisos del Usuario Recepcionista	77
3.13. Permisos del Usuario Instructor	77
3.14. Menú de Membresías	78
3.15. Menú de Clientes	79

3.16. Interfaz para Tomar Fotografías	80
3.17. Menú de Pagos	81
3.18. Menú de Ejercicios	82
3.19. Menú de Rutinas	83
3.20. Interfaz del perfil Rutina	84
3.21. Menú de Historiales	85
3.22. Perfil del Cliente	86
3.23. Historial del Cliente	87
3.24. Asignar Etiqueta al Cliente	87
3.25. Identificación de Cliente Activo	88
3.26. Identificación de Cliente Activo en GymID	89
3.27. Identificación de Cliente Inactivo	90
3.28. Identificación de Cliente Inactivo en GymID	90
C.1. Dimensiones de la cubierta del Lector de etiquetas NFC	136

GLOSARIO

ISO 14443: es una norma internacional relacionada con las tarjetas de identificación electrónicas. Esta norma define una tarjeta de proximidad utilizada para identificación y pagos. La norma consta de cuatro partes: características físicas, potencia de radiofrecuencia e interfaz de señales, Inicialización y anticolidión, y Protocolo de transmisión.

MIFARE: es una tecnología de tarjetas inteligentes sin contacto, de las más ampliamente instaladas en el mundo, es equivalente a las 3 primeras partes de la norma ISO 14443 Tipo A de 13.56 MHz con protocolo de alto nivel, con una distancia típica de lectura de 10 cm.

NFC: son las siglas del término inglés que significa comunicación de campo cercano, esta es una tecnología de comunicación inalámbrica de corto alcance y alta frecuencia que permite el intercambio de información entre dispositivos.

RFID: son las siglas del término inglés que significa identificación por radiofrecuencia, esta es una tecnología de identificación remota e inalámbrica en la cual un dispositivo lector se comunica a través de una antena con etiquetas mediante ondas de radio.

SISTEMA RFID: utiliza un lector con un microcontrolador incrustado y una antena que opera a 13,56 MHz. El lector mantiene a su alrededor un campo electromagnético de modo que al acercarse una tarjeta al campo, esta se alimenta eléctricamente de esta energía inducida y puede establecerse la comunicación lector-tarjeta.

UML: son las siglas del término inglés que significa Lenguaje Unificado de Modelado, este es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y compuestos reciclados.

RESUMEN

La implementación de nuevas tecnologías ha acelerado a grandes pasos el desarrollo tecnológico mundial, NFC no es una tecnología relativamente nueva, sin embargo, con la implementación en teléfonos móviles y la necesidad de fortalecer la seguridad y velocidad de los sistemas de identificación, la tecnología NFC ha vuelto a tener protagonismo¹.

Este trabajo plantea el diseño e implementación de un sistema de gestión para un gimnasio utilizando la tecnología NFC que permite optimizar el redimiendo de los servicios ofrecidos por un gimnasio a un bajo costo, dando la posibilidad a pequeñas empresas de competir con las grandes. La implementación del sistema ofrece una novedosa forma de gestionar la entrada de clientes al gimnasio, permitiendo al cliente identificarse por medio de tarjetas NFC que mejorará significativamente la velocidad de acceso al gimnasio. Adicionalmente el sistema incluye una aplicación para escritorio que cuenta con una completa lista de herramientas que administran toda la información de clientes, pagos, membresías, rutinas y ejercicios.

El documento está dividido en tres partes, en el primero se analiza los requerimientos de hardware y software para el desarrollo del proyecto, el segundo presenta el diseño del dispositivo lector de etiquetas NFC y de la aplicación para escritorio, por último se muestra el montaje del dispositivo y el funcionamiento principal de la aplicación. Durante todo el documento se utilizan los diagramas de UML para modelar las partes correspondientes al análisis, diseño e implementación de la aplicación para escritorio.

PALABRAS CLAVE: NFC. Radio frecuencia. Sistema de identificación. Administración de Gimnasio. Banco de datos. Software libre.

¹PEÑA, Carlos. ¿Qué es NFC y para qué sirve? [en línea]. En: Paréntesis. Junio. 2015. Disponible en Internet: http://www.parentesis.com/tutoriales/Que_es_NFC_y_para_que_sirve

INTRODUCCIÓN

Las nuevas tecnologías ayudan a las empresas a optimizar sus funcionalidades; incrementando sus ingresos, mejorando los procesos e implementando nuevas herramientas. En Colombia las empresas buscan estar a la vanguardia de las tecnologías utilizadas en países desarrollados, también se incentiva el conocimiento en semilleros de investigación para reducir costos de adquisición y desarrollar tecnologías inexistentes.

Un sistema de gestión es una serie de actividades coordinadas que se utilizan para realizar la administración en una empresa. Para ello se desarrolla software especializado que permite gestionar bases de datos, planificación, procedimientos, toma de decisiones, y cada una de las funciones propias de una empresa y de sus empleados².

Los gimnasios han ido incursionando en la implementación de nuevas tecnologías que les permiten optimizar su gestión administrativa. La mayoría han incorporado software computacionales para almacenar la información de los clientes en bases de datos, consiguiendo que el administrador del gimnasio maneje la información de manera organizada; mejorando significativamente su funcionamiento y minimizando pérdidas económicas por errores humanos. Sin embargo, en muchos casos la información que se almacena en las bases de datos no es suficiente, las aplicaciones informáticas solamente guardan datos de relevancia de los clientes y pagos, limitando la gestión administrativa. Además, la incorporación de dispositivos eléctricos que complementen las aplicaciones informáticas aún son insuficientes.

NFC es una tecnología de comunicación inalámbrica, de corto alcance y alta frecuencia que permite el intercambio de datos entre dispositivos, se caracteriza principalmente por su velocidad de comunicación al ser casi instantánea ya que no requiere emparejamiento previo. A pesar de ser una extensión de la tecnología RFID, se diferencia por ser de alcance muy reducido, lo que la vuelve una tecnología más segura³. Esta tecnología fue aprobada en el 2003 como estándar ISO, sin embargo se ha dado a conocer recién

²Qué son los sistemas de gestión empresarial? [en línea]. Buenos Aires: Informática hoy. Disponible en Internet: <http://www.informatica-hoy.com.ar/aprender-informatica/Que-son-los-sistemas-de-gestion-empresarial.php>

³Diferencia entre NFC y RFID [en línea]. En: Mundo NFC. Febrero, 2012. Disponible en Internet: <https://mundonfc.wordpress.com/2012/02/08/diferencia-entre-nfc-y-rfid/>

temente en implementaciones para teléfonos celulares⁴. También ha ido reemplazando a la tecnología RFID y hoy la podemos encontrar en el transporte público y tarjetas de crédito.

El sistema de gestión desarrollado para el gimnasio es un conjunto de componentes de software y hardware que utiliza la tecnología NFC para mejorar la velocidad de autenticación de los clientes. El chip NFC se encuentra integrado en las tarjetas, manillas y llaveros que son entregados a los clientes del gimnasio para que puedan utilizarlos al momento de su identificación; estos cuentan con un número único de identificación (ID) que al ser acercados al lector NFC, adquirirá su ID para enviarlo por medio de una conexión serial al computador.

El lector NFC se construye por medio de la plataforma de hardware libre Arduino que adhiere un módulo de lectura y escritura para etiquetas NFC. Se realiza un algoritmo de programación para Arduino que sea capaz de configurar y activar la lectura de etiquetas, enviar y recibir datos del computador, y escribir información de relevancia para el cliente en la pantalla LCD.

GYMID es la aplicación para escritorio utilizada para el sistema de gestión, creada a partir del entorno de desarrollo Qt Creator que utiliza lenguaje de programación C++. La aplicación no solo permite administrar la información de los clientes en bases de datos, sino que cumple con una variedad de funcionalidades que le posibilitan al gimnasio tener información completa de clientes, pagos, membresías, ejercicios y rutinas de entrenamiento.

GYMID ofrece nuevas características para la gestión de un gimnasio, cómo la posibilidad de implementar una modalidad de cobro que permite a los clientes pagar por días asistidos en vez de periodos fijos. También ofrece versatilidad de ejercicios ya que permite a los instructores agregar ejercicios no convencionales o propios del gimnasio a la base de datos, para luego ser utilizados en rutinas de entrenamiento. Además es una aplicación segura, cada usuario del programa deberá estar registrado y solo podrá ingresar cuando inicie sesión, los usuarios están divididos por categorías que definen el control de la administración. Igualmente crea informes para el usuario que muestran historiales de los pagos realizados, historiales de visitas hechas por clientes y registros de actividades de los usuarios del programa.

⁴ PASTOR, Javier. Siete cosas que puedes hacer gracias a la tecnología NFC [en línea]. En: Xataka Móvil. Marzo, 2013. Disponible en Internet: <http://www.xatakamovil.com/futuro/7-cosas-que-puedes-hacer-gracias-a-la-tecnologia-nfc>

1. ANÁLISIS DE REQUERIMIENTOS

1.1. PLANTEAMIENTO DEL PROBLEMA

Actualmente en la mayoría de gimnasios la gestión administrativa se realiza de forma manual, las inscripciones de los clientes se escriben en papel, la autenticación la realiza el administrador del gimnasio pidiendo la cédula de ciudadanía o nombre de los clientes y verificando los registros de papel, no existe un sistema que tenga información completa sobre los pagos realizados, deudas, datos personales de clientes y rutinas de entrenamiento.

La verificación de identidad y la inscripción al gimnasio se realizan de manera lenta, el archivar los datos en papel limita la posibilidad de tener la información al instante, no existen copias de las rutinas de entrenamiento, provocando que la gestión sea ineficiente.

La inexistencia de un control en la entrada al gimnasio posibilita que los clientes entren en mora fácilmente ya que la cancelación del servicio recae considerablemente en la voluntad del mismo, esto permite que personas inescrupulosas aprovechen la falla para seguir utilizando el servicio cuando su periodo de pago ha expirado y finalmente esto se traduce en pérdidas económicas para el gimnasio.

Algunos gimnasios cuentan con programas informáticos para la gestión administrativa y bases de datos en medios virtuales, organizando mejor la información de los clientes. Sin embargo estos programas suelen ser costosos y la información almacenada en la base de datos es insuficiente. El acceso continúa siendo un problema, realizándose con la identificación por cédula de ciudadanía, causando mayor congestión en la entrada del gimnasio y retardando el ingreso.

1.2. PLANTEAMIENTO DEL CASO

Se requiere crear un sistema que pueda leer tarjetas de identificación, obtener la ID (Identificación NFC) y enviarla al computador, por medio de un software de escritorio

se hará una consulta a la base de datos la cual muestre la información del cliente en una ventana de perfil, donde el recepcionista podrá verificar la autenticidad del cliente por medio fotográfico y el estado de activación de la cuenta, si el cliente tiene pagos activos o disponibles, además de un historial de pagos. El perfil muestra datos específicos del cliente como nombres, apellidos, documento de identidad, si trabaja o estudia, además de datos de interés personales como tipo de sangre, fecha de nacimiento, etc.

Al momento de leer las tarjetas, el sistema deberá mostrar al cliente información esencial como el estado de la cuenta del cliente, la fecha de inicio, la fecha final y los días que le faltan para que termine su activación. Además, el sistema deberá llevar un historial de visitas de cada cliente con la respectiva fecha y hora del momento de su identificación.

La aplicación de software debe suplir la necesidades de un sistema de gestión de un gimnasio, administrando la información de clientes, pagos y membresías. Además, deberá cumplir con funciones especiales para la administración de rutinas de ejercicios para cada cliente con la opción de suministrar ejercicios propios del gimnasio.

La aplicación deberá ser capaz de crear diferentes usuarios para su administración con variantes en su interfaz dependiendo del tipo de usuario (Recepcionista, Instructor, Administrador), cada acción del usuario identificado deberá ser almacenada llevando así un historial de actividades. También deberá cumplir con funciones que suplan las necesidades del sistema, como tomar fotografías por medio de una cámara web instalada para la identificación de los clientes o generar documentos en archivos PDF imprimibles para las rutinas creadas.

1.3. ACTORES

Los actores son los roles jugados por usuarios del sistema, estos representan actividades que son relevantes a las especificaciones de los casos de uso asociados. La Figura 1.1 presenta los diferentes actores del sistema, además se especifica gráficamente la relación de herencia existente entre los mismos.

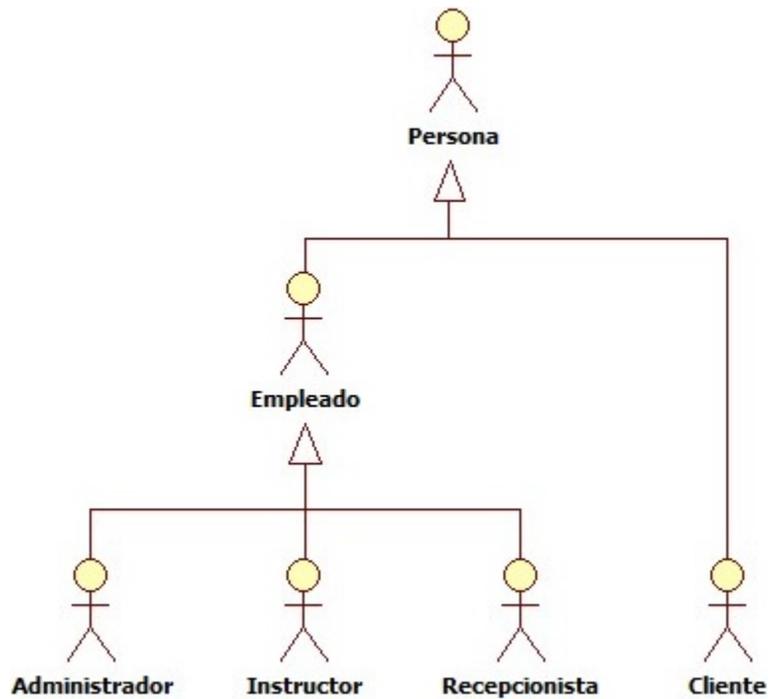


Figura 1.1: Diagrama de herencia de los Actores

Administrador: Persona encargada de Administrar el gimnasio. Otorga permiso a usuarios del sistema y puede acceder a todas las funciones del sistema.

Recepcionista: Persona encargada de supervisar y atender en la entrada a los clientes. Administra opciones relacionadas con clientes, pagos y funciones básicas del sistema.

Instructor: Persona encargada de entrenar al cliente. Administra opciones relacionadas con ejercicios y rutinas.

Cliente: Persona suscrita al gimnasio. Receptora del servicio, mantiene funciones limitadas en el sistema y correspondientes a su cuenta.

Los Actores Administrador, Instructor y Recepcionista heredan características de la clase Empleado ya que estos trabajan para el gimnasio y tienen atributos como salario, identificaciones, ocupación, etc. Adicionalmente todos los actores del sistema tienen una clase padre en común Persona y heredan atributos como nombre, apellido, fecha de nacimiento, etc.

1.4. DIAGRAMAS DE CASOS DE USO

En esta sección se presentan los casos de uso para cada actor en diagramas de casos de uso. Un caso de uso es una descripción de las acciones de un sistema desde el punto de vista del actor. Esta es una herramienta importante que permite analizar los requerimientos del sistema desde el punto de vista de cada actor.

1.4.1 Diagrama de casos de uso del sistema. El diagrama está compuesto por 9 casos de uso principales que resumen el funcionamiento del sistema, la Figura 1.2 muestra los casos de uso resultantes presentados para todos los actores del sistema.

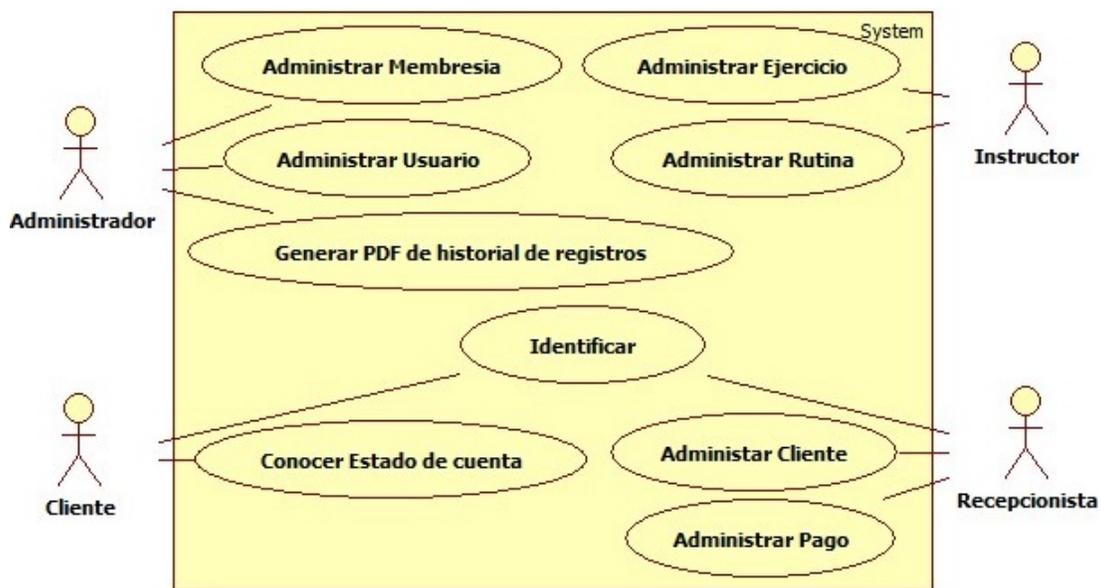


Figura 1.2: Diagrama de Casos de uso del Sistema

El caso de uso Identificar es compartido por los actores Cliente y Recepcionista ya que ambos actores se benefician del mismo, permitiendo al Cliente identificarse ante el sistema y al Recepcionista Identificar a los Clientes del sistema. Para más información sobre el caso de uso, revisar el Cuadro A.10 del Anexo A Especificaciones de los Casos de Uso.

1.4.2 Diagrama de casos de uso para el Administrador. El diagrama está compuesto por 3 casos de uso principales que utiliza el actor Administrador, 3

1.4. DIAGRAMAS DE CASOS DE USO

casos de uso (Crear Membresía, Editar Membresía y Eliminar Membresía) son extensiones del caso de uso Administrar Membresía y 2 casos de uso (Crear Usuario y Eliminar Usuario) son extensiones del caso de uso Administrar Usuario, conformando un total de 8 casos de uso que hacen parte del sistema visto por el actor Administrador. La Figura 1.3 muestra los casos de uso presentados para el actor Administrador del gimnasio.

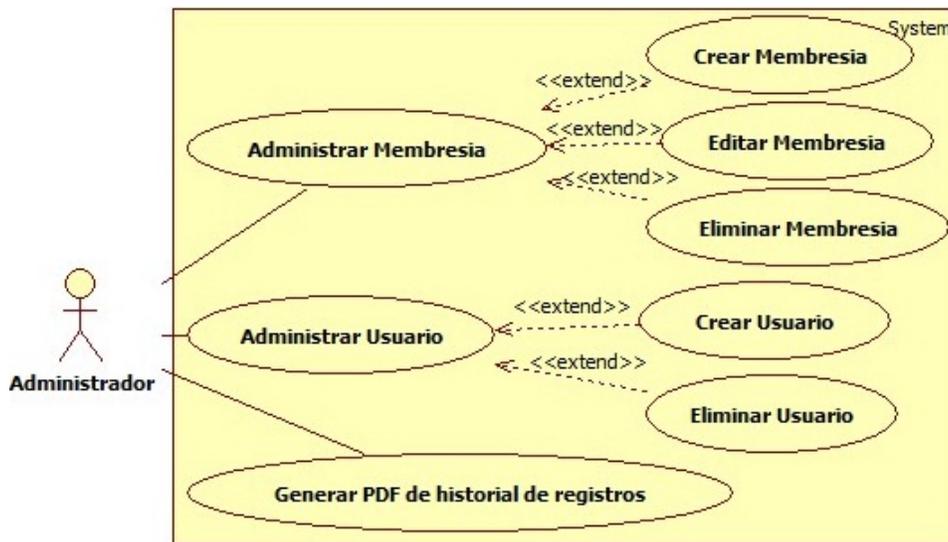


Figura 1.3: Diagrama de Casos de uso para el Administrador

El análisis de cada caso de uso se encuentra descrito en la Sección A.1 del Anexo A Especificaciones de los Casos de Uso.

1.4.3 Diagrama de casos de uso para el Cliente. El diagrama está compuesto por 2 casos de uso principales que utiliza el actor Cliente del gimnasio, la Figura 1.4 muestra los casos de uso presentados para el actor Cliente del gimnasio.



Figura 1.4: Diagrama de Casos de uso para el Cliente

El análisis de cada caso de uso se encuentra descrito en la Sección A.2 del Anexo A Especificaciones de los Casos de Uso.

1.4.4 Diagrama de casos de uso para el Instructor. El diagrama está compuesto por 2 casos de uso principales que utiliza el actor Instructor, 5 casos de uso (Crear Ejercicio, Editar Ejercicio, Eliminar Ejercicio, Importar Ejercicio y Exportar Ejercicio) son extensiones del caso de uso Administrar Ejercicio y 4 casos de uso (Crear Rutina, Editar Rutina, Eliminar Rutina y Generar PDF de rutinas) son extensiones del caso de uso Administrar Rutina, conformando un total de 11 casos de uso que hacen parte del sistema visto por el actor Instructor. La Figura 1.5 muestra los casos de uso presentados para el actor Instructor del gimnasio.

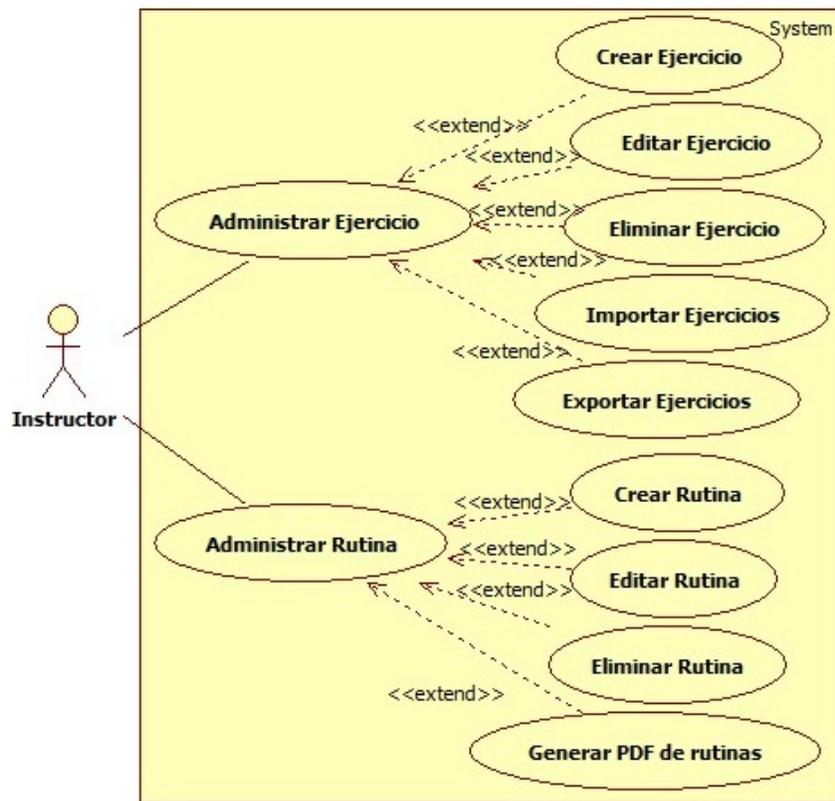


Figura 1.5: Diagrama de Casos de uso para el Instructor

El análisis de cada caso de uso se encuentra descrito en la Sección A.3 del Anexo A Especificaciones de los Casos de Uso.

1.4.5 Diagrama de casos de uso para el Recepcionista. El diagrama está compuesto por 3 casos de uso principales que utiliza el actor Recepcionista, 4 casos

1.5. ALCANCES

de uso (Crear Cliente, Editar Cliente, Eliminar Cliente y Asignar Identificaciones) son extensiones del caso de uso Administrar Cliente, de los cuales 2 de estos casos de uso (Crear Cliente y Editar Cliente) utilizan un mismo caso de uso Tomar Fotografías y 3 casos de uso (Crear Pago, Eliminar Pago y Pagar Saldo) son extensiones del caso de uso Administrar Pago, conformando un total de 11 casos de uso que hacen parte del sistema visto por el actor Recepcionista. La Figura 1.6 muestra los casos de uso presentados para el actor Recepcionista del gimnasio.

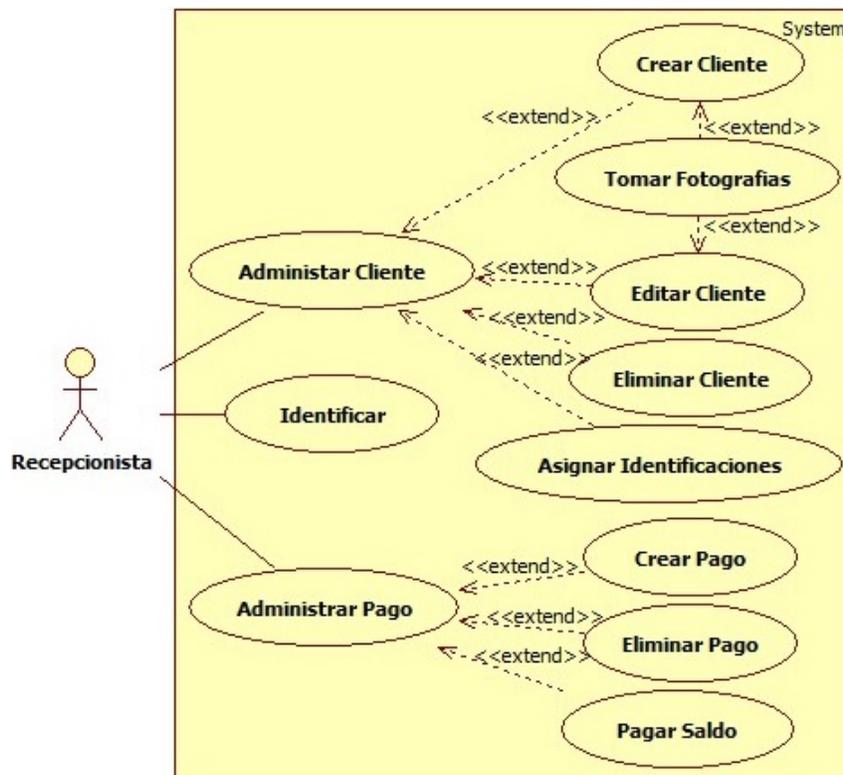


Figura 1.6: Diagrama de Casos de uso para el Recepcionista

El análisis de cada caso de uso se encuentra descrito en la Sección A.4 del Anexo A Especificaciones de los Casos de Uso.

1.5. ALCANCES

La implementación del proyecto dará una imagen moderna y sofisticada al gimnasio alcanzando los siguientes aspectos:

- El recepcionista encargado podrá identificar a los clientes del gimnasio de manera

rápida y confiable incrementando así la calidad del servicio.

- El programa brindará la posibilidad de tomar fotografías correspondientes a los clientes, provocando un aumento en la seguridad al evitar la suplantación de identidades.
- El programa presentará la opción de acceder a la información del cliente por medio del número de identificación en caso de que las etiquetas NFC se extravíen o dañen.
- El programa tendrá una base de datos en la cual se almacenarán los registros de cada cliente de forma organizada, optimizando la gestión de la información.
- El gimnasio tendrá la opción de ofrecer un nuevo servicio de inscripción prepagada que le permitirá pagar por los días que asiste, aumentando y mejorando los servicios ofrecidos.

1.6. LIMITACIONES

La implementación del proyecto estará limitada en los siguientes aspectos:

- El sistema está sujeto a posibles errores humanos en cuanto a la manipulación del software, excepto en la funcionalidad del mismo.
- El precio de adquisición de las etiquetas NFC está sujeto al lugar donde se realice la compra de las mismas, al ser una tecnología de vanguardia es probable que se requiera buscar de alternativas extranjeras.
- El sistema sería vulnerable ante la suplantación de identidad realizada por hermanos gemelos, ya que se trata de un método de identificación indirecta.

1.7. ARQUITECTURA DEL SISTEMA

Para diseñar el sistema se requiere la elección de los componentes adecuados para el diseño del hardware y software, cada uno se desarrolla paralelamente ya que son complementarias, la primera se refiere al diseño físico del sistema y la segunda al diseño virtual del sistema.

1.7.1 Arquitectura del hardware. Para el diseño físico del sistema se ha encontrado como mejor solución la utilización de hardware libre Arduino y los escudos compatibles para Arduino. Además del Arduino y el escudo, se agrega un circuito que configure la pantalla LCD y el LED de estados, este empaquetado de elementos se conoce como sistema de lectura de etiquetas NFC. Los dispositivos y elementos electrónicos fueron elegidos teniendo en cuenta la compatibilidad, los costos, la optimización y la accesibilidad.

1.7.1.1 Etiquetas NFC. Teniendo en cuenta el estándar internacional ISO/IEC 14443 que trata sobre circuitos integrados en tarjetas sin contacto también conocidas como tarjetas de proximidad o identificación, que define los tipos de tarjetas y sus protocolos de operación, las cuales se diferencian principalmente por los protocolos de inicialización, los métodos de modulación y codificación, se eligen las etiquetas MIFARE de ISO/IEC 14443 de tipo A principalmente por su fácil accesibilidad y bajo costo en el mercado. La Figura 1.7 muestra las etiquetas NFC Mifare.



Figura 1.7: Etiquetas NFC Mifare

Fuente: Enzocard Research and Development. Mifare Card.

1.7.1.2 Escudo lector de etiquetas NFC. El escudo debe ser un dispositivo integrado al Arduino que pueda leer las ID de las etiquetas NFC, además debe cumplir con funciones como escritura de información en la memoria EEPROM de la etiqueta, se elige el escudo Adafruit PN532 ya que cumple con las necesidades especificadas y puede comunicarse bajo el bus serial I2C minimizando los pines utilizados. Este dispositivo utiliza 1 I/O digital y 2 entradas análogas. La Figura 1.8 muestra el escudo Adafruit PN532.

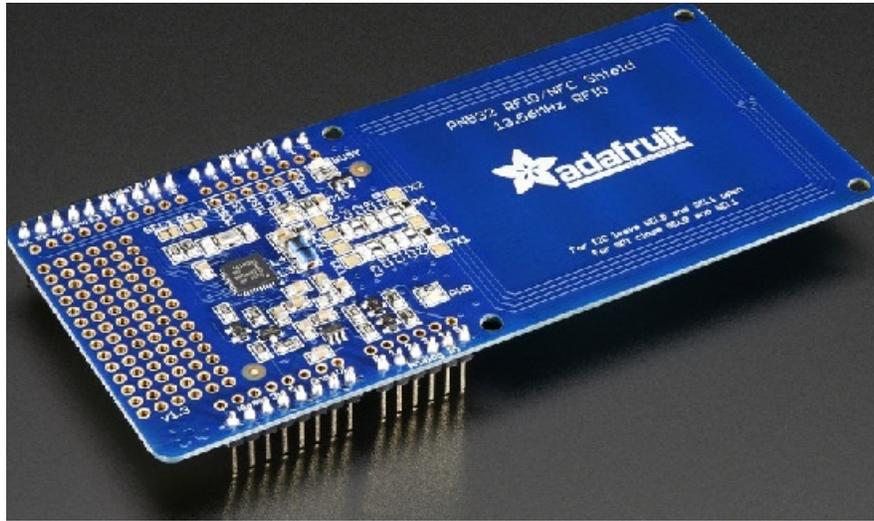


Figura 1.8: Escudo Adafruit PN532

Fuente: Adafruit. Adafruit PN532 NFC/RFID Controller Shield for Arduino + Extras.

Este escudo está basado en el microcontrolador PN532 que es un excelente módulo de transmisión integrado para comunicación sin contacto a 13.56 MHz, soporta comunicación sin contacto de dispositivos MIFARE entre otras, utilizando velocidades de transferencia de hasta 424 kbit/s en ambas direcciones. El PN532 implementa un sistema de demodulación y decodificación de señales de los dispositivos ISO/IEC 14443A/MIFARE. El microcontrolador cuenta con una gran variedad de modos de operación, para este proyecto el modo más apropiado es utilizar el PN532 como Lector/Escritor de ISO/IEC 14443A/MIFARE, este modo permite la lectura y escritura de etiquetas Mifare. La Figura 1.9 presenta la comunicación entre el PN532 en modo Lector/Escritor de ISO/IEC 14443A/MIFARE y los dispositivos MIFARE.

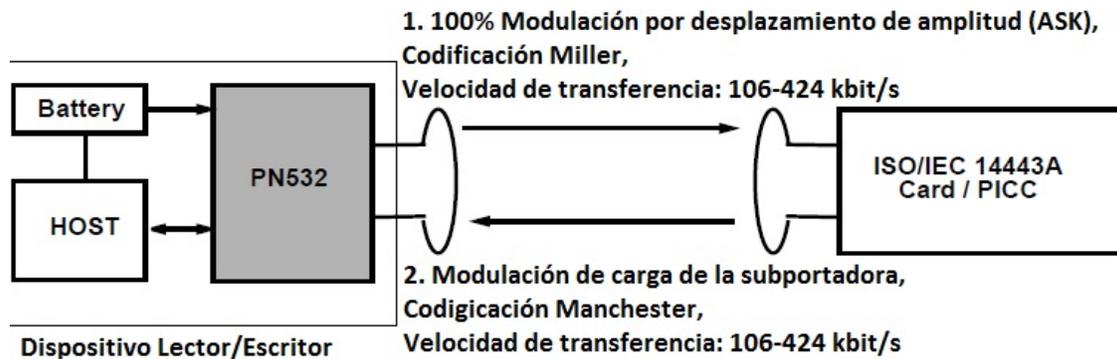


Figura 1.9: Comunicación entre PN532 y Etiqueta Mifare

Fuente: NXP Semiconductors. PN532/C1: Near Field Communication (NFC) controller (Traducido al español). Rev 3,2. 2007. 224 p.

1.7. ARQUITECTURA DEL SISTEMA

El PN532 no solo requiere de una fuente de alimentación y una antena, sino que también requiere de un 'Host' que lo controle, para este caso se usa el microcontrolador ATmega328P que integra el Arduino UNO que cumple con los requerimientos, ya que soporta comunicación I2C, tiene entradas análogas y convertidores Análogo-Digitales.

1.7.1.3 Pantalla LCD. Para dar una mejor visualización de la información se elige la pantalla LCD 4x20, dando la posibilidad de mostrar 4 líneas de texto al tiempo. Este dispositivo utiliza 6 I/O digitales bajo el modo de operación de 4-bits. La Figura 1.10 muestra la pantalla LCD 20x4.

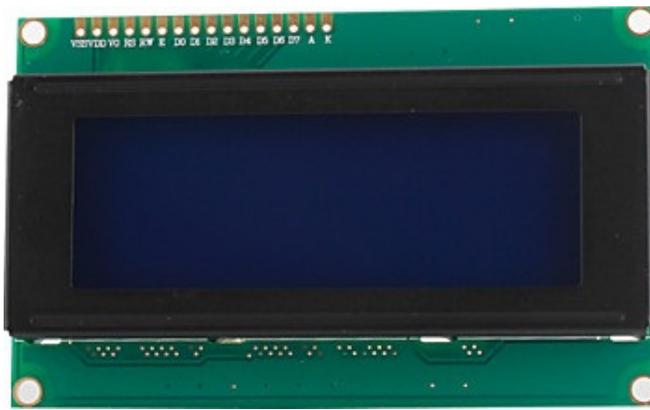


Figura 1.10: Pantalla LCD

Fuente: Didacticas Electrónicas. Display LCD caracteres 4x20.

1.7.1.4 LED. El sistema ocupa dos estado en la lectura de etiquetas. El primer estado es el 'Estado de Espera' que se encuentra activo mientras no se coloque ninguna etiqueta en el lector. El segundo estado es el 'Estado Ocupado' que se activa después de que el lector detecte una etiqueta. El LED que puede mostrar el mejor comportamiento del dispositivo de lectura de etiquetas NFC es LED Bicolor (RG, GB, RB, etc.), minimizando el espacio en el montaje y dando la posibilidad de mostrar más de 2 estados del dispositivo. Este elemento utiliza 2 I/O digitales. La Figura 1.11 muestra el LED bicolor de 5mm de alto brillo.



Figura 1.11: LED Bicolor

Fuente: Cetronic. LED BICOLOR ROJO Y AMARILLO 5mm L59EYW.

1.7.1.5 Arduino. Considerando en número de entradas y salidas de datos que necesitan los dispositivos adheridos al Arduino, da como resultado 9 I/O Digitales y 2 I Análogas en total, sin tener en cuenta los pines en común como las tierras (GND) y los voltajes de alimentación (5V). Debido a estos resultados se elige el Arduino UNO R3, una placa basada en el microcontrolador ATmega328P que cuenta con 6 pines I análogas y 14 pines I/O digitales, suficientes para soportar el sistema. Adicionalmente Arduino incluye el microcontrolador ATmega16U2 que establece una interfaz de comunicación serial entre el circuito integrado ATmega328P y el ordenador por medio del bus estándar USB. La Figura 1.11 muestra el Arduino UNO R3.



Figura 1.12: Arduino UNO R3

Fuente: Idea Digital. Arduino UNO.

1.7.2 Arquitectura del software. Para el diseño virtual del sistema se opta por software libre, como entorno de desarrollo de aplicaciones para escritorio se elige Qt Creator que utiliza lenguaje de programación C++, además MySQL como sistema gestor de bases de datos.

1.7.2.1 Lenguaje de Programación. Se utiliza C++ como lenguaje de programación para desarrollar el software por su rapidez, eficiencia, librerías estándar y la capacidad de permitir programación estructurada y orientada a objetos al mismo tiempo.

1.7.2.2 Qt Creator. La aplicación de escritorio requiere de interfaz gráfica, comunicación con el puerto serial, comunicación con la base de datos y administración de formularios. Qt Creator permite realizar estas funcionalidades de manera eficiente en las aplicaciones creadas por medio de sus librerías integradas.

1.7.2.3 MySQL. El sistema requiere de una adquisición veloz de la información en la base de datos y una capacidad de gestionar bases de datos de gran tamaño de manera eficiente. MySQL es un gestor de base de datos rápido y potente, optimizando la velocidad de las operaciones realizadas en bases de datos de gran tamaño. Además MySQL es fiable y seguro en el manejo de la información, su bajo consumo de recursos informáticos permite que la base de datos pueda trabajar en cualquier máquina.

2. DISEÑO

2.1. CLASES DEL SISTEMA

En esta sección se presentan las clases del sistema en diagramas de clases, primero se realiza un recorrido por los actores y sus herencias de clases padres que ayuda a un mejor entendimiento del rol de cada actor, luego se presenta el diagrama de clases del sistema que muestra el sistema completo, sus interacciones y un diccionario de atributos y operaciones utilizados en cada clase.

2.1.1 Diagrama de herencia en actores. El diagrama muestra la herencia de atributos y operaciones para cada clase actora del diagrama de clases, los atributos muestran la información detallada de una clase y las operaciones son las actividades que una clase puede realizar. Cada atributo y operación tiene una visibilidad dentro de la clase, pueden ser pública (+), privada (-) o protegida (#) lo cual mostrará que puede ser visto o utilizado por otras clases. La Figura 2.1 presenta el diagrama de herencia en las clases de actores del sistema: Cliente, Recepcionista, Administrador e Instructor.

Las clases Recepcionista, Administrador e Instructor son empleados del gimnasio, por lo tanto heredan atributos de la clase Empleado como el Cargo que desempeña dentro del gimnasio, este atributo no solo lo heredan las clases hijas, sino que también es visible para cualquier clase y un ID del Gimnasio que refiere a un código o número de identificación dentro del gimnasio en el que trabajan. También heredan la operación Atender() que se refiere al trabajo independiente que tiene cada empleado con los clientes, teniendo en cuenta el cargo que desempeñan.

Todas las clases de actores heredan atributos de la clase Persona que se encuentra en el tope de la jerarquía, estos atributos se refieren a la información personal que tiene cada ser humano dentro de una sociedad. Los Nombres, Apellidos, DNI, Genero, Fecha de nacimiento, Tipo de sangre, Numero de Teléfono y Ocupación son atributos que son útiles para el análisis y requisitos para el correcto funcionamiento del sistema.

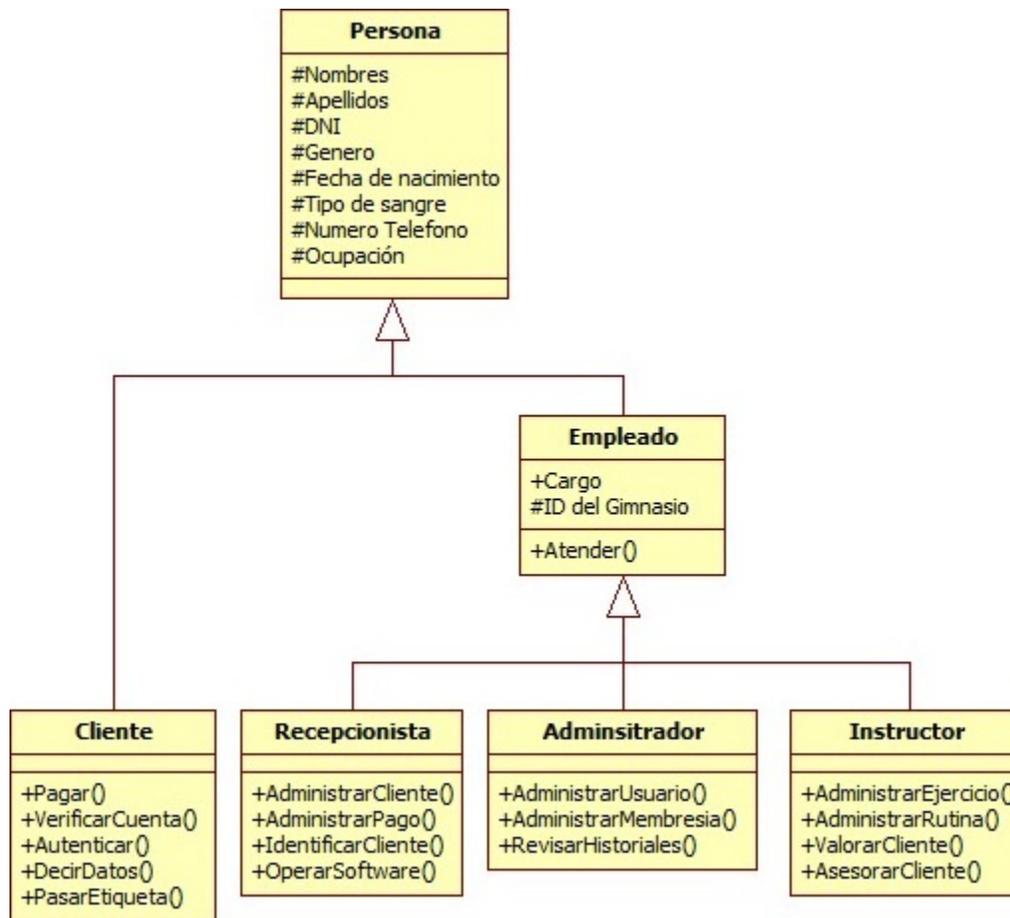


Figura 2.1: Diagrama de Herencia en Actores del Sistema

2.1.2 Diagrama de clases. El diagrama de clases muestra la estructura estática del sistema, mostrando las clases, sus atributos, operaciones y relaciones. En cada clase se encuentran los atributos y operaciones más importantes, además las clases están relacionadas por medio de asociaciones que son unidireccionales (cuando una clase utiliza a otra) o son bidireccionales (cuando existe mutuo beneficio). La Figura 2.2 presenta el diagrama de clases del sistema.

Para empezar el análisis de la estructura, empezamos por analizar la clase **Recepcionista** que al ser un actor del sistema, en la Sección 2.1.1 se explicó el rol que cumple y los atributos que heredan de clases padres, así que solo nos limitaremos en analizar las operaciones que las clases actoras ejecutan.

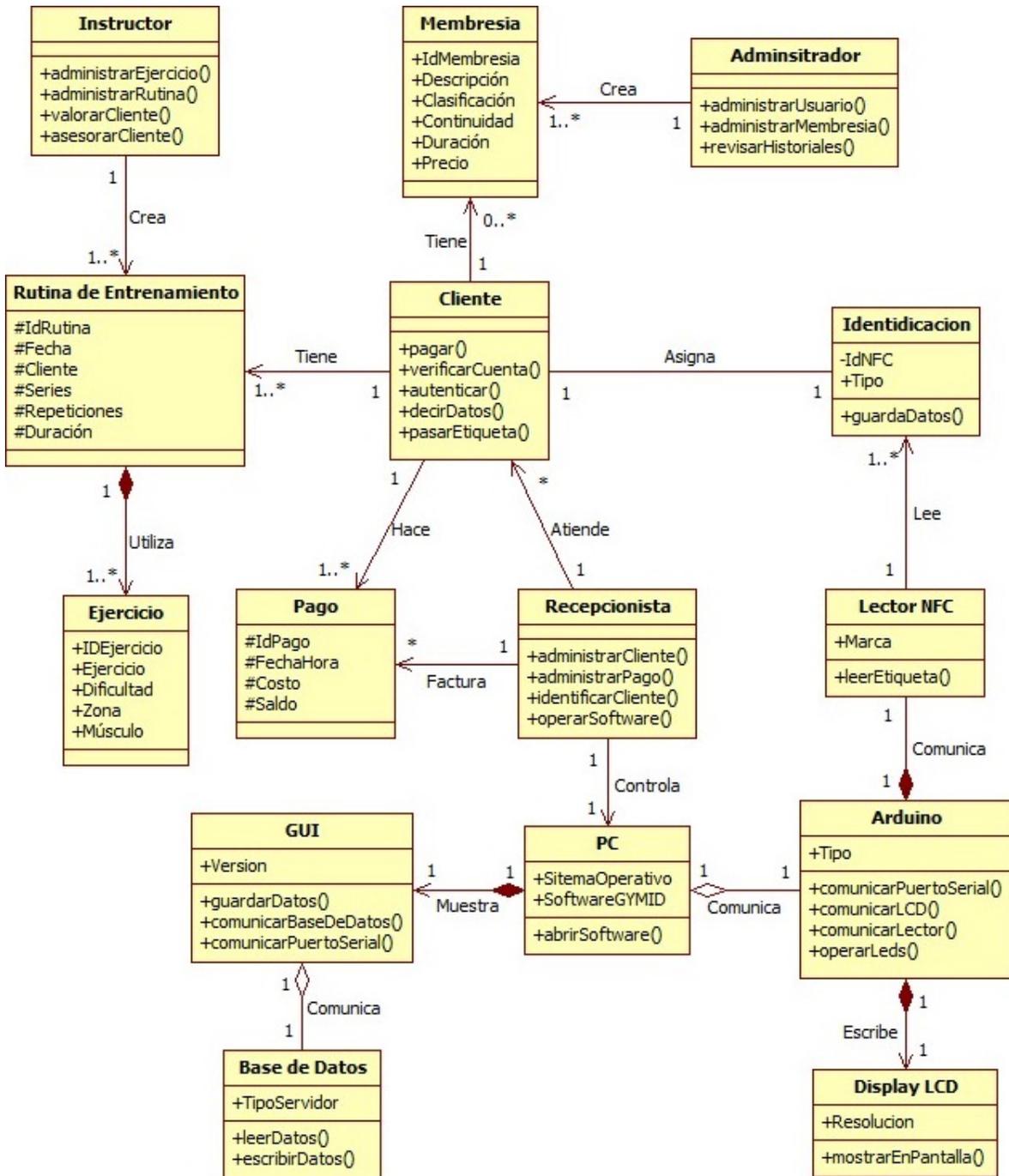


Figura 2.2: Diagrama de Clases del Sistema

2.1.3 Clase Recepcionista. El Recepcionista se asocia en 3 oportunidades, en todas las ocasiones un Recepcionista atiende a muchos Clientes, factura muchos Pagos y controla un PC.

2.1.3.1 Operaciones.

- `administrarCliente()`: cuando se atiende al Cliente para crear, editar o eliminar datos del cliente en la base de datos, esto lo realiza por medio del PC.
- `administrarPago()`: cuando se atiende al Cliente para crear, editar o eliminar datos de pagos del cliente en la base de datos, esto lo realiza por medio del PC.
- `identificarCliente()`: es la acción de identificación de cada Cliente en el momento de ingreso al gimnasio.
- `operarSoftware()`: se refiere al manejo del software en el PC, el Recepcionista debe saber cómo funciona el software GYMID.

2.1.4 Clase Pago. El Pago se asocia con las clases Cliente y Recepcionista, sin embargo el Pago no es una clase base en asociaciones, así que solo es receptor de asociaciones de otras clases.

2.1.4.1 Atributos.

- `IdPago`: es el código del pago al momento de registrarlo.
- `FechaHora`: es la fecha y la hora del registro del pago.
- `Costo`: es la cantidad que el cliente debe pagar.
- `Saldo`: es la cantidad que el cliente paga.

2.1.5 Clase PC. El PC se asocia en 3 oportunidades, en dos de las ocasiones un PC muestra una GUI y comunica un Arduino, la tercera asociación ocurre entre el Recepcionista y el PC como receptor. En la relación de composición con la GUI, el PC se construye a partir de la GUI, mientras en la relación de agregación con el Arduino, el PC utiliza al Arduino para su funcionamiento en el sistema.

2.1.5.1 Atributos.

- `SistemaOperativo`: sistema operativo del PC, debe tener uno preestablecido para que el software funcione adecuadamente.
- `SoftwareGYMID`: aplicación para escritorio GYMID.

2.1.5.2 Operaciones.

- `abrirSoftware()`: se refiere a la acción de ejecutar la aplicación GYMID para que el Recepcionista pueda empezar a trabajar.

2.1.6 Clase GUI. La GUI que se refiere a la interfaz gráfica del software GYMID, se asocia en 2 oportunidades, en una de las ocasiones una GUI comunica una Base de Datos, la segunda asociación ocurre entre el PC y la GUI como receptor. En la relación de agregación con la Base de Datos, la GUI utiliza a la Base de Datos para su funcionamiento en el sistema.

2.1.6.1 Atributos.

- `Versión`: versión de la aplicación GYMID.

2.1.6.2 Operaciones.

- `guardarDatos()`: guarda en la base de datos las acciones que se realicen en la GUI.
- `comunicarBaseDeDatos()`: comunica el software GYMID con la base de datos.
- `comunicarPuertoSerial()`: comunica el software GYMID con el puerto serial del computador (esto establece la comunicación con el Arduino).

2.1.7 Clase Base de Datos. La Base de Datos se asocia con la clase GUI, al ser una relación bidireccional, existe una mutua comunicación entre ambas clases.

2.1.7.1 Atributos.

- `TipoServidor`: es el servidor donde va a trabajar la base de datos.

2.1.7.2 Operaciones.

- `leerDatos()`: operación de lectura de datos (dentro de la base de datos).
- `escribirDatos()`: operación de escritura de datos (dentro de la base de datos).

2.1.8 Clase Arduino. El Arduino se asocia en 3 oportunidades, en dos de las ocasiones un Arduino escribe en un Display LCD y comunica con un Lector NFC, la tercera asociación ocurre entre el PC y el Arduino, al ser una relación bidireccional existe una mutua comunicación entre ambas clases. La clase Arduino esta compuesta por 2 clases adicionales que son Display LCD y Lector NFC, este conjunto de clases es un subsistema que refiere al hardware de lectura de etiquetas NFC, esto significa que el Arduino se construye a partir del Display LCD y el Lector NFC.

2.1.8.1 Atributos.

- Tipo: modelo del producto Arduino, que se elige dependiendo de las necesidades del sistema.

2.1.8.2 Operaciones.

- `comunicarPuertoSerial()`: comunica el Arduino con el computador.
- `comunicarLCD()`: comunica el Arduino con el display LCD.
- `comunicarLector()`: comunica el Arduino con el lector NFC.
- `operarLeds()`: controla Leds que serán útiles para la visualización de estados del sistema de lectura de etiquetas NFC.

2.1.9 Clase Display LCD. El Display LCD se asocia con la clase Arduino, sin embargo el Display LCD no es una clase base en asociaciones, así que solo es receptor de la asociación.

2.1.9.1 Atributos.

- Resolución: dimensión horizontal y vertical expresadas en píxeles del display LCD.

2.1.9.2 Operaciones.

- `mostrarEnPantalla()`: escribe en pantalla.

2.1.10 Clase Lector NFC. El Lector NFC se asocia en 2 oportunidades, en una ocasión un Lector NFC lee una o muchas Identificaciones, en la segunda relación el Lector NFC se asocia con la clase Arduino, al ser una relación bidireccional, existe una mutua comunicación entre ambas clases.

2.1.10.1 Atributos.

- Marca: tipo de producto.

2.1.10.2 Operaciones.

- leerEtiqueta(): detecta etiqueta NFC y obtiene su ID.

2.1.11 Clase Identificación. La Identificación que se refiere al objeto con etiqueta NFC se asocia en 2 oportunidades, en la primera ocasión una Identificación se le asigna a un Cliente, la segunda asociación ocurre entre el Lector NFC y la Identificación como receptor.

2.1.11.1 Atributos.

- IdNFC: número de identificación de etiquetas NFC.
- Tipo: tipo de objeto con etiqueta.

2.1.11.2 Operaciones.

- guardaDatos(): almacena datos en etiquetas con memoria EEPROM.

2.1.12 Clase Cliente. El Cliente se asocia en 5 oportunidades, en tres ocasiones un Cliente hace uno o muchos Pagos, tiene una o muchas Rutinas de Entrenamiento y tiene ninguna o muchas Membresías, la cuarta asociación ocurre entre el Recepcionista y Cliente como receptor y la quinta asociación ocurre entre el Cliente y la Identificación que al ser bidireccional, un Cliente también se le asigna una Identificación.

2.1.12.1 Operaciones.

- pagar(): realiza pagos al recepcionista.
- verificarCuenta(): conoce el estado de la cuenta por medio del Display LCD.
- autenticar(): se identifica ante el Recepcionista.
- decirDatos(): dice sus datos personales, especificado en sus atributos (Ver Figura 2.1 Diagrama de Herencia en Actores del Sistema).
- pasarEtiqueta(): acción de pasar la etiqueta NFC por el lector. Esta operación es utilizada en varias ocasiones durante el desarrollo del sistema.

2.1.13 Clase Instructor. El Instructor se asocia en 1 oportunidad, en esa ocasión un Instructor crea una o muchas Rutinas de Entrenamiento.

2.1.13.1 Operaciones.

- administrarEjercicio(): el instructor crea, edita o elimina ejercicios en la base de datos, esto lo realiza por medio del PC.
- administrarRutina(): el instructor crea, edita o elimina rutinas de entrenamiento en la base de datos, esto lo realiza por medio del PC.
- valorarCliente(): califica los músculos del cliente para poder crear rutinas de entrenamiento.
- asesorarCliente(): atiende el cliente y lo ayuda en sus entrenamientos.

2.1.14 Clase Rutina de Entrenamiento. La Rutina de Entrenamiento se asocia en 3 oportunidades, en una de las ocasiones una Rutina de Entrenamiento utiliza uno o muchos Ejercicios, la segunda asociación ocurre entre el Instructor y la Rutina de Entrenamiento como receptor y la tercera asociación ocurre entre el Cliente y la Rutina de Entrenamiento como receptor. En la relación de composición con el Ejercicio, la Rutina de Entrenamiento se construye a partir de los Ejercicios.

2.1.14.1 Atributos.

- IdRutina: código de la rutina, se especifica al momento de crearla.
- Fecha: fecha de la rutina de entrenamiento.
- Cliente: nombre del cliente al que se le asigna la rutina.
- Series: cantidad de veces que debe hacer las repeticiones.
- Repeticiones: cantidad veces que debe repetir el ejercicio.
- Duración: duración en tiempo del ejercicio.

2.1.15 Clase Ejercicio. El Ejercicio se asocia con la clase Rutina de Entrenamiento, sin embargo el Ejercicio no es una clase base en asociaciones, así que solo es receptor de la asociación.

2.1.15.1 Atributos.

- IDEjercicio: código del ejercicio, se especifica al momento de crearlo.
- Ejercicio: nombre del ejercicio.
- Dificultad: dificultad para realizar el ejercicio.
- Zona: lugar del cuerpo donde tiene efecto el ejercicio.
- Musculo: musculo involucrado con el ejercicio.

2.1.16 Clase Administrador. El Administrador se asocia en 1 oportunidad, en esa ocasión un Administrador crea una o muchas Membresías.

2.1.16.1 Operaciones.

- administrarUsuario(): el administrador crea o elimina usuarios del software en la base de datos, esto lo realiza por medio del PC.
- administrarMembresia(): el administrador crea, edita o elimina membresías en la base de datos, esto lo realiza por medio del PC.
- revisarHistoriales(): revisa historiales de la base de datos, esto lo realiza por medio del PC.

2.1.17 Clase Membresía. La Membresía se asocia con las clases Cliente y Administrador, sin embargo la Membresía no es una clase base en asociaciones, así que solo es receptor de asociaciones de otras clases.

2.1.17.1 Atributos.

- IdMembresía: código de la membresía, se especifica al momento de crearlo.
- Descripción: nombre de la membresía, el nombre puede ser escrito como una descripción.
- Clasificación: tipo de membresía (estudiante, particular, etc.).
- Continuidad: tipo de cobro (por días o periodos de tiempo).
- Duración: cantidad de duración de la membresía en días.
- Precio: costo de la membresía.

2.2. DIAGRAMAS DE SECUENCIAS

En esta sección se presentan los diagramas de secuencias para cada caso de uso. Cada diagrama de secuencias muestra la interacción entre actores, objetos e interfaces gráficas del sistema durante el tiempo.

2.2.1 Administrar Membresía. El caso de uso Administrar Membresía perteneciente al actor Administrador muestra la interacción entre el Administrador, las interfaces de membresías y la base de datos. La Figura 2.3 presenta el diagrama de secuencias del caso de uso Administrar Membresía.

El caso de uso inicia cuando el Administrador del gimnasio abre la GUI de Membresías, esta muestra al Administrador las opciones que puede realizar.

Si el Administrador elige Crear Membresía, GUI de Membresías abre la GUI de Crear Membresía y esta es mostrada al Administrador. El Administrador llena los datos de la nueva membresía y oprime el botón Crear, si algunos datos no fueron completados en los campos, GUI de Crear Membresía informa al Administrador del error; si todos los datos de los campos fueron completados, GUI de Crear Membresía crea la membresía en la Base de Datos, cierra la ventana e informa al Administrador que la membresía fue creada.

Si el Administrador elige Editar Membresía, GUI de Membresías abre la GUI de Editar Membresía y esta es mostrada al Administrador. El Administrador escribe el código de la membresía y GUI de Editar Membresía busca la membresía en la Base de Datos, si la membresía no existe en la Base de Datos, GUI de Editar Membresía informa al Administrador; si la membresía si existe, GUI de Editar Membresía carga los datos de la membresía y la muestra al Administrador. El Administrador edita los datos y oprime el botón Guardar, GUI de Editar Membresía guarda los cambios de la membresía en la Base de Datos, cierra la ventana e informa al Administrador que la membresía fue editada.

Si el Administrador elige Eliminar Membresía, GUI de Membresías abre la GUI de Eliminar Membresía y esta es mostrada al Administrador. El Administrador selecciona la membresía y oprime el botón Eliminar, si la membresía no fue seleccionada, GUI de Eliminar Membresía informa al Administrador del error; si la membresía fue seleccionada, GUI de Eliminar Membresía elimina la membresía en la Base de Datos, cierra la ventana e informa al Administrador que la membresía fue eliminada.

2.2. DIAGRAMAS DE SECUENCIAS

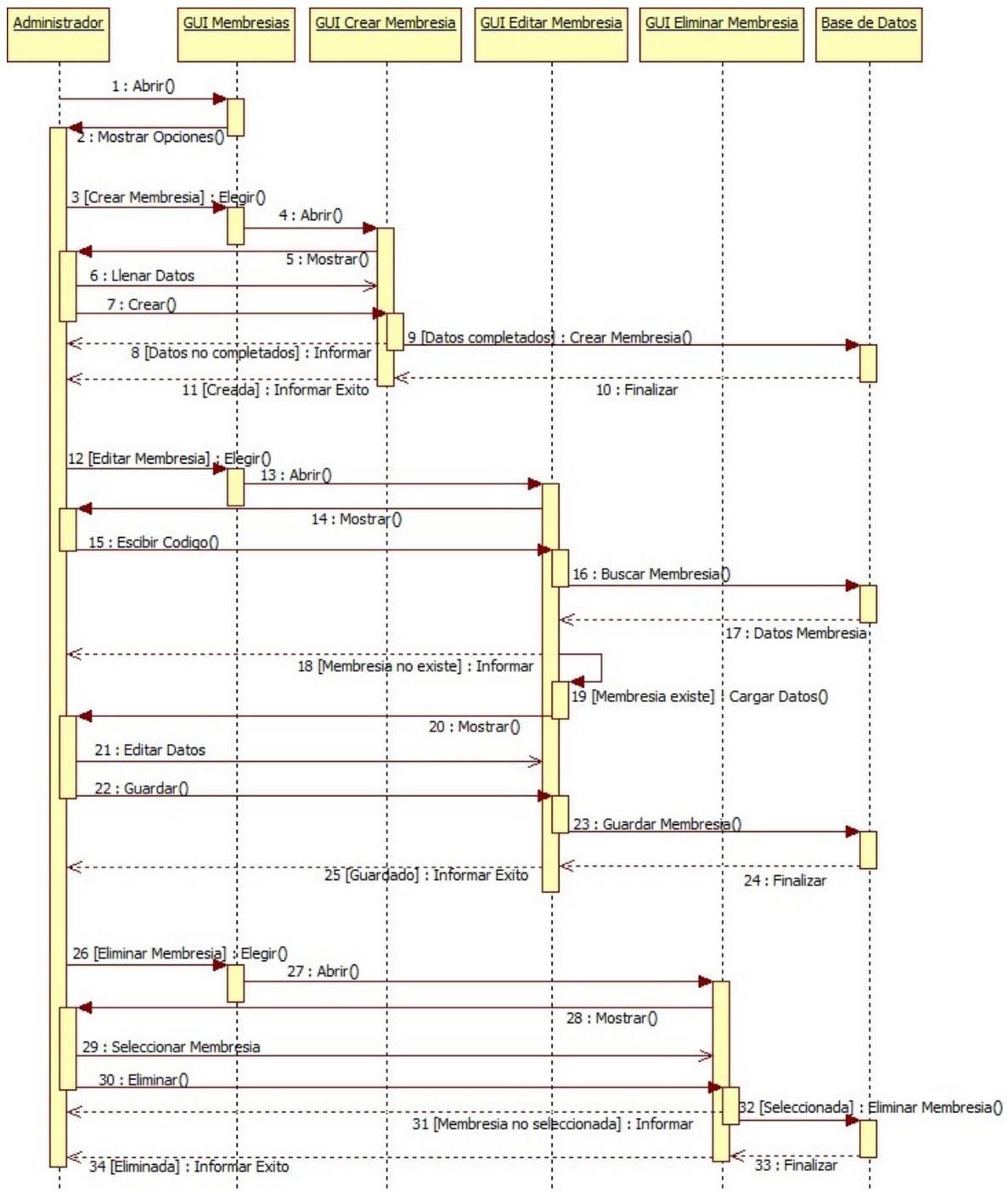


Figura 2.3: Diagrama Secuencias para Administrar Membresía

2.2.2 Administrar Usuario. El caso de uso Administrar Usuario pertenece al actor Administrador muestra interacción entre el Administrador, las interfaces

de usuarios y la base de datos. La Figura 2.4 presenta el diagrama de secuencias del caso de uso Administrar Usuario.

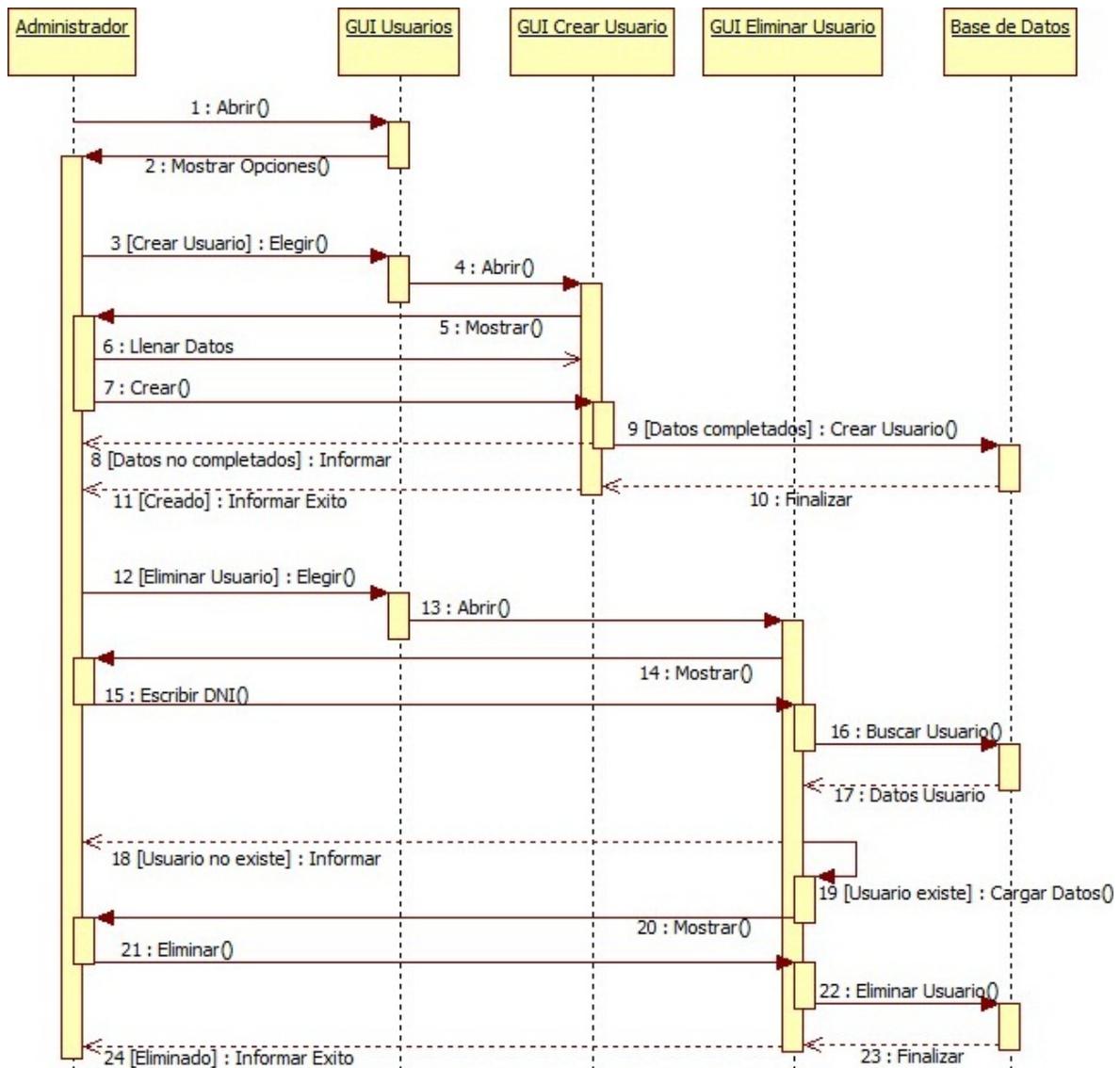


Figura 2.4: Diagrama Secuencias para Administrar Usuario

El caso de uso inicia cuando el Administrador del gimnasio abre la GUI de Usuarios, esta muestra al Administrador las opciones que puede realizar.

Si el Administrador elige Crear Usuario, GUI de Usuarios abre la GUI de Crear Usuario y esta es mostrada al Administrador. El Administrador llena los datos del nuevo usuario y oprime el botón Crear, si algunos datos no fueron completados en los campos, GUI de Crear Usuario informa al Administrador del error; si todos los datos

2.2. DIAGRAMAS DE SECUENCIAS

de los campos fueron completados, GUI de Crear Usuario crea el usuario en la Base de Datos, cierra la ventana e informa al Administrador que el usuario fue creado.

Si el Administrador elige Eliminar Usuario, GUI de Usuarios abre la GUI de Eliminar Usuario y esta es mostrada al Administrador. El Administrador escribe el número de DNI del usuario y GUI de Eliminar Usuario busca el usuario en la Base de Datos, si el usuario no existe en la Base de Datos, GUI de Eliminar Usuario informa al Administrador; si el usuario si existe, GUI de Eliminar Usuario carga los datos del usuario y muestra al Administrador. El Administrador oprime el botón Eliminar, GUI de Eliminar Usuario elimina el usuario en la Base de Datos, cierra la ventana e informa al Administrador que el usuario fue eliminado.

2.2.3 Generar PDF de historial de registros. El caso de uso Generar PDF de Historial de registros perteneciente al actor Administrador muestra la interacción entre el Administrador, la interfaz de historiales, la base de datos y el PDF de historiales. La Figura 2.5 presenta el diagrama de secuencias del caso de uso Generar PDF de Historial de registros.

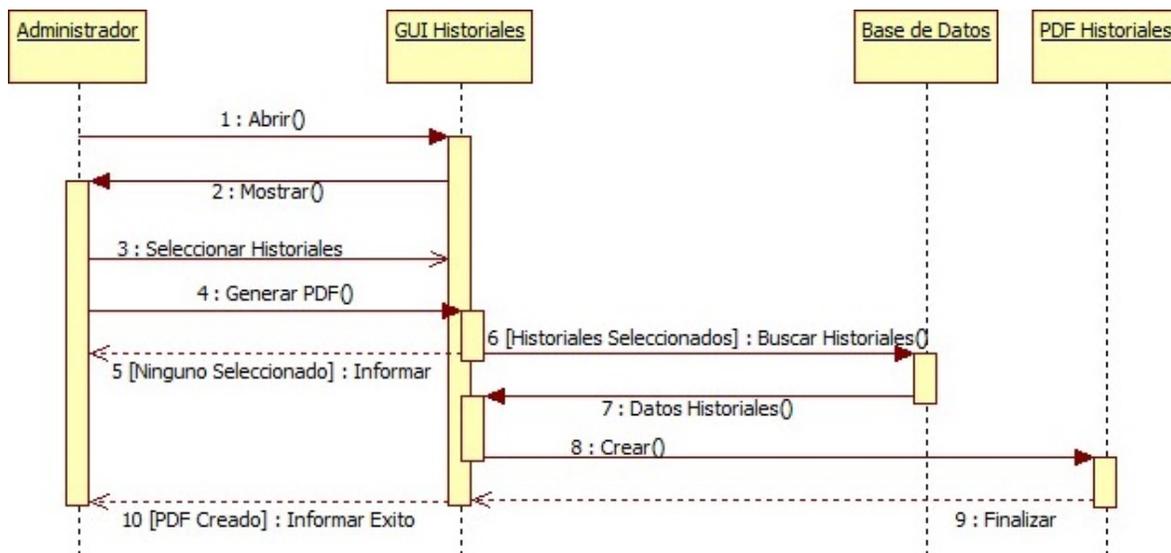


Figura 2.5: Diagrama Secuencias para Generar PDF Historial de Registros

El caso de uso inicia cuando el Administrador del gimnasio abre la GUI de Historiales y esta es mostrada. El Administrador selecciona los historiales que desea imprimir y oprime el botón Generar PDF, si ningún historial fue seleccionado, GUI de Historiales informa al Administrador del error; de lo contrario GUI de Historiales busca los historiales en la Base de Datos, los almacena y crea el PDF de Historiales en la ubicación especificada, luego GUI de Historiales cierra la ventana e informa al Administrador que el PDF de Historiales fue creado.

2.2.4 Identificar. El caso de uso Identificar perteneciente a los actores Cliente y Recepcionista muestra la interacción entre el Cliente, el Recepcionista, la interfaz gráfica principal, la base de datos, el lector NFC y el display LCD. La Figura 2.6 presenta el diagrama de secuencias del caso de uso Identificar.

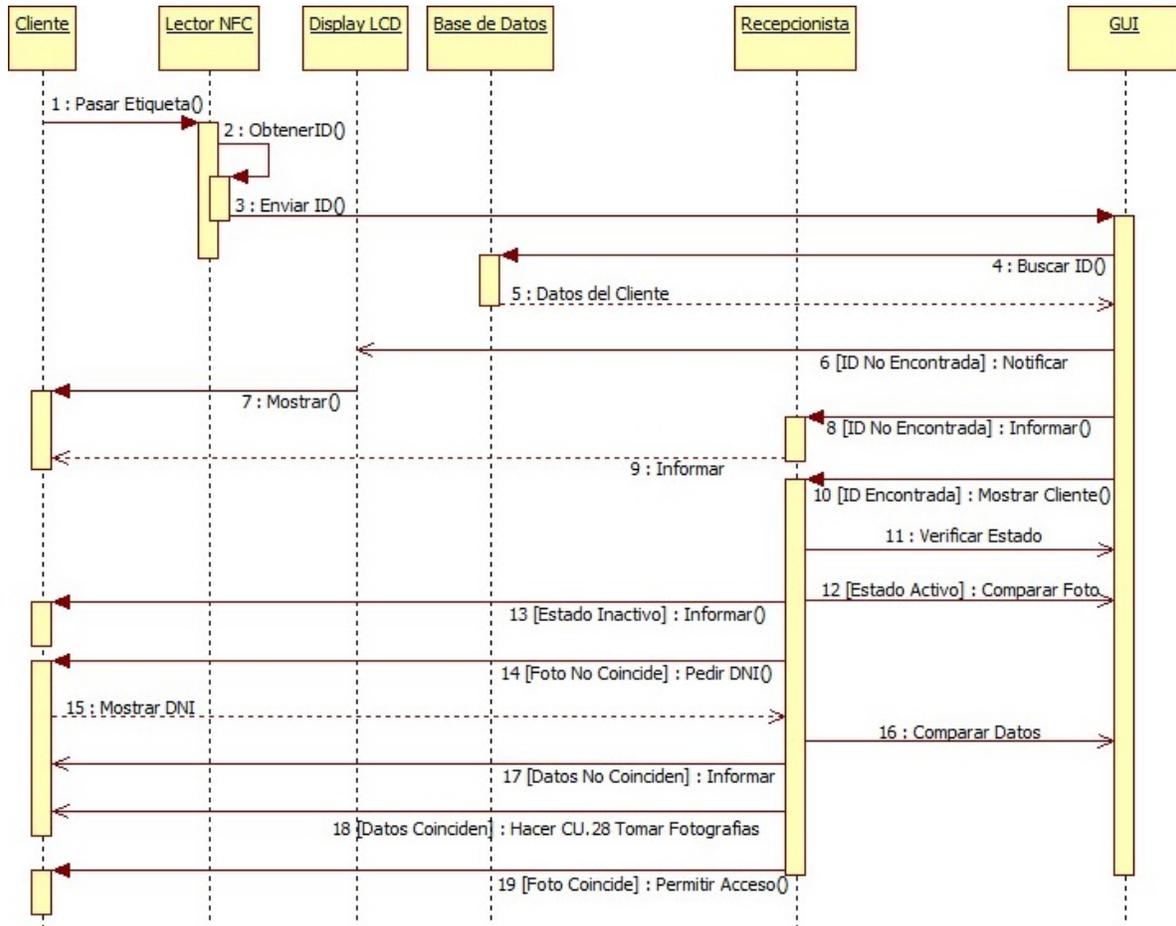


Figura 2.6: Diagrama Secuencias para Identificar

El caso de uso inicia cuando el Cliente del gimnasio pasa la etiqueta por el Lector NFC. El Lector obtiene la ID de la etiqueta y la envía a la GUI Principal, la GUI busca la ID en la base de datos y recibe la información del cliente, si la ID no fue encontrada, GUI escribe en el Display LCD que el Cliente no fue encontrado e informa al Recepcionista el error; de lo contrario GUI muestra el perfil del Cliente. El Recepcionista verifica el estado del Cliente, si el estado es Inactivo, el Recepcionista informa al Cliente; si el estado es Activo, el Recepcionista Compara la foto del perfil con el Cliente; si la foto coincide, el Recepcionista permite el acceso del Cliente; si la foto no coincide, el Recepcionista pide el DNI al Cliente y compara los datos; si los datos no coinciden, el Recepcionista informa al Cliente e impide el acceso del Cliente; si los datos coinciden

2.2. DIAGRAMAS DE SECUENCIAS

el Recepcionista toma una nueva fotografía al Cliente (ver CU.28 Tomar Fotografías) y permite el acceso del Cliente.

2.2.5 Conocer estado de cuenta. El caso de uso Conocer Estado de Cuenta perteneciente al actor Cliente muestra la interacción entre el Cliente, el Recepcionista, la interfaz gráfica principal, la base de datos, el lector NFC y el display LCD. La Figura 2.7 presenta el diagrama de secuencias del caso de uso Conocer Estado de Cuenta.

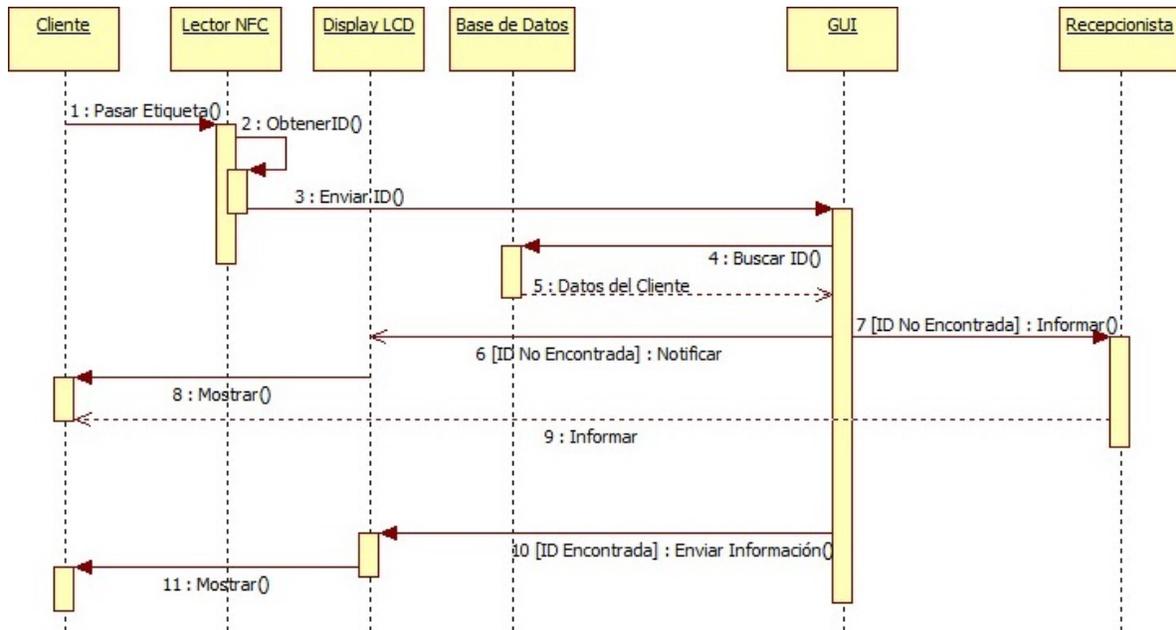


Figura 2.7: Diagrama Secuencias para Conocer Estado de Cuenta

El caso de uso inicia cuando el Cliente del gimnasio pasa la etiqueta por el Lector NFC. El Lector obtiene la ID de la etiqueta y la envía a la GUI Principal, la GUI busca la ID en la base de datos y recibe la información del Cliente, si la ID no fue encontrada, GUI escribe en el Display LCD que el Cliente no fue encontrado e informa al Recepcionista el error; si la ID fue encontrada, GUI escribe en el Display LCD información importante de la cuenta (estado de la cuenta, fecha de inicio de la membresía, fecha final de la membresía y días restantes) y esta es mostrada al Cliente.

2.2.6 Administrar Ejercicio. El caso de uso Administrar Ejercicio perteneciente al actor Instructor muestra la interacción entre el Instructor, las interfaces de ejercicios y la Base de Datos. La Figura 2.8 presenta el diagrama de secuencias del caso de uso Administrar Ejercicio.

El caso de uso inicia cuando el Instructor del gimnasio abre la GUI de Ejercicios, esta muestra al Instructor las opciones que puede realizar. Si el Instructor elige Crear Ejercicio, GUI de Ejercicios abre la GUI de Crear Ejercicio y esta es mostrada al Instructor. El Instructor llena los datos del nuevo ejercicio y oprime el botón Crear, si algunos datos no fueron completados en los campos, GUI de Crear Ejercicio informa al Instructor del error; si todos los datos de los campos fueron completados, GUI de Crear Ejercicio crea el ejercicio en la Base de Datos, cierra la ventana e informa al Instructor que el ejercicio fue creado.

Si el Instructor elige Editar Ejercicio, GUI de Ejercicios abre la GUI de Editar Ejercicio y esta es mostrada al Instructor, El Instructor escribe el código del ejercicio y GUI de Editar Ejercicio busca el ejercicio en la Base de Datos, si el ejercicio no existe en la Base de Datos, GUI de Editar Ejercicio informa al Instructor; si el ejercicio si existe, GUI de Editar Ejercicio carga los datos del ejercicio y muestra al Instructor. El Instructor edita los datos y oprime el botón Guardar, GUI de Editar Ejercicio guarda los cambios del ejercicio en la Base de Datos, cierra la ventana e informa al Instructor que el ejercicio fue editado.

Si el Instructor elige Eliminar Ejercicio, GUI de Ejercicios abre la GUI de Eliminar Ejercicio y esta es mostrada al Instructor. El Instructor selecciona el ejercicio y oprime el botón Eliminar, si el ejercicio no fue seleccionado, GUI de Eliminar Ejercicio informa al Instructor del error; si el ejercicio fue seleccionado, GUI de Eliminar Ejercicio elimina el ejercicio en la Base de Datos, cierra la ventana e informa al Instructor que el ejercicio fue eliminado.

Si el Instructor elige Importar Ejercicios, GUI de Ejercicios abre la GUI de Importar Ejercicios y esta es mostrada al Instructor. El Instructor especifica la ubicación donde se encuentra el archivo (.txt) con los ejercicios y oprime el botón Importar, si la ubicación no fue especificada, GUI de Importar Ejercicios informa al Instructor del error; si la ubicación fue especificada, GUI de Importar Ejercicios guarda los ejercicios del archivo (.txt) en la Base de Datos, cierra la ventana e informa al Instructor que los ejercicios fueron importados.

Si el Instructor elige Exportar Ejercicios, GUI de Ejercicios abre la GUI de Exportar Ejercicios y esta es mostrada al Instructor. El Instructor especifica la ubicación donde quiere que se guarden los ejercicios y oprime el botón Exportar, si la ubicación no fue especificada, GUI de Exportar Ejercicios informa al Instructor del error; si la ubicación fue especificada, GUI de Exportar Ejercicios guarda los ejercicios de la Base de Datos en un archivo (.txt) en la ubicación especificada, cierra la ventana e informa al Instructor que los ejercicios fueron exportados.

2.2. DIAGRAMAS DE SECUENCIAS

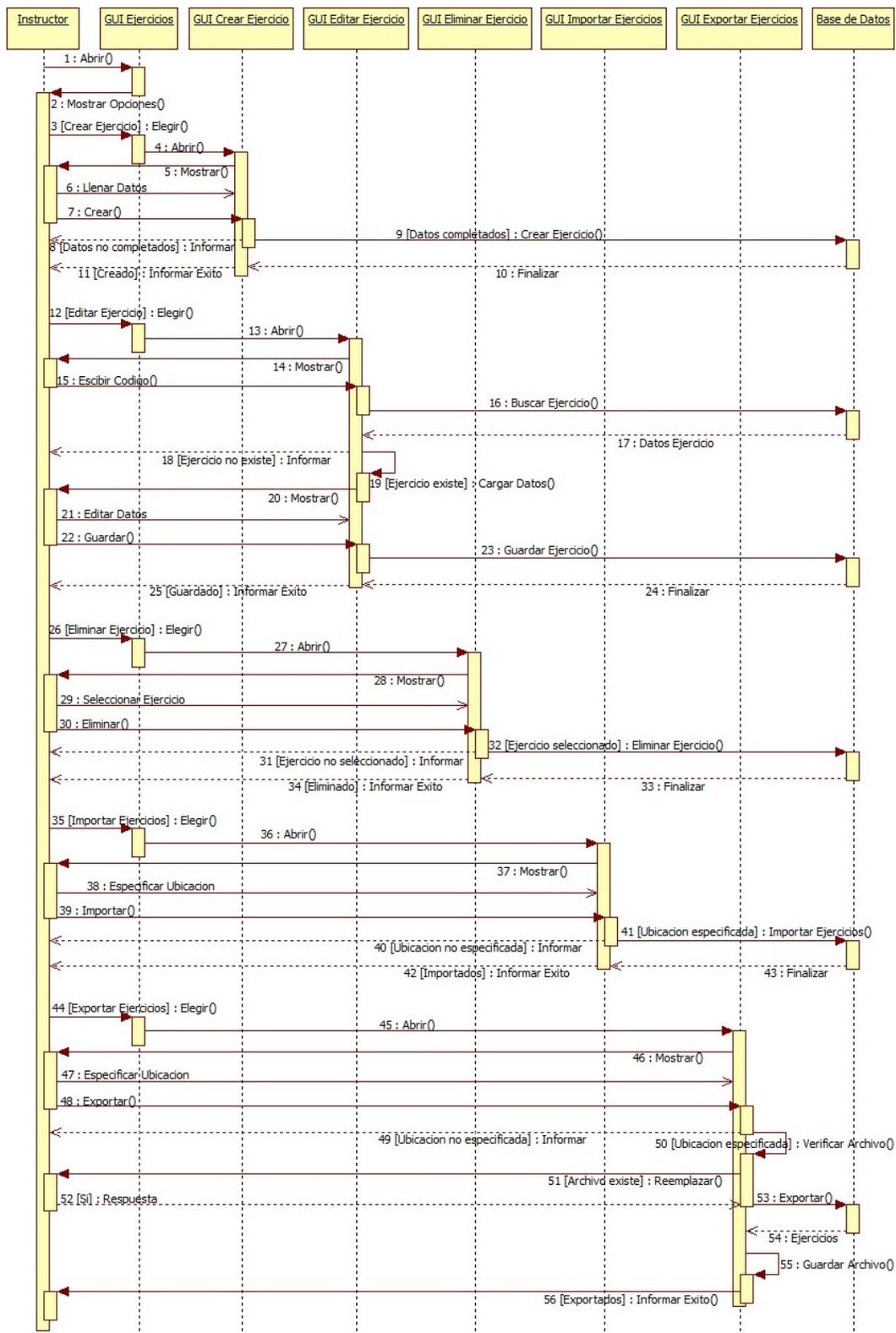


Figura 2.8: Diagrama Secuencias para Administrar Ejercicio

2.2.7 Administrar Rutina. El caso de uso Administrar Rutina perteneciente al actor Instructor muestra la interacción entre el Instructor, las interfaces de rutinas, la base de datos y el PDF de rutinas. La Figura 2.9 presenta el diagrama de secuencias del caso de uso Administrar Rutina.

El caso de uso inicia cuando el Instructor del gimnasio abre la GUI de Rutinas, esta muestra al Instructor las opciones que puede realizar.

Si el Instructor elige Crear Rutina, GUI de Rutinas abre la GUI de Crear Rutina y esta es mostrada al Instructor. El Instructor llena los datos de la nueva rutina y oprime el botón Crear, si algunos datos no fueron completados en los campos, GUI de Crear Rutina informa al Instructor del error; si todos los datos de los campos fueron completados, GUI de Crear Rutina crea la rutina en la Base de Datos, cierra la ventana e informa al Instructor que la rutina fue creada.

Si el Instructor elige Editar Rutina, GUI de Rutinas abre la GUI de Editar Rutina y esta es mostrada al Instructor. El Instructor escribe el código de la rutina y GUI de Editar Rutina busca la rutina en la Base de Datos, si la rutina no existe en la Base de Datos, GUI de Editar Rutina informa al Instructor; si la rutina si existe, GUI de Editar Rutina carga los datos de la rutina y muestra al Instructor. El Instructor edita los datos y oprime el botón Guardar, GUI de Editar Rutina guarda los cambios de la rutina en la Base de Datos, cierra la ventana e informa al Instructor que la rutina fue editada.

Si el Instructor elige Eliminar Rutina, GUI de Rutinas abre la GUI de Eliminar Rutina y esta es mostrada al Instructor. El Instructor selecciona la rutina y oprime el botón Eliminar, si la rutina no fue seleccionada, GUI de Eliminar Rutina informa al Instructor del error; si la rutina fue seleccionada, GUI de Eliminar Rutina elimina la rutina en la Base de Datos, cierra la ventana e informa al Instructor que la rutina fue eliminada.

Si el Instructor elige Ver Listado, GUI de Rutinas muestra el listado de las rutinas creadas. El Instructor busca la rutina que quiere ver y la abre, GUI de Rutinas busca la rutina en la Base de Datos, la carga en la ventana y la muestra al Instructor. El Instructor oprime el botón Generar PDF y GUI de Rutinas crea un archivo PDF con la rutina del cliente organizada por días, luego abre el archivo PDF y lo muestra al Instructor.

2.2. DIAGRAMAS DE SECUENCIAS

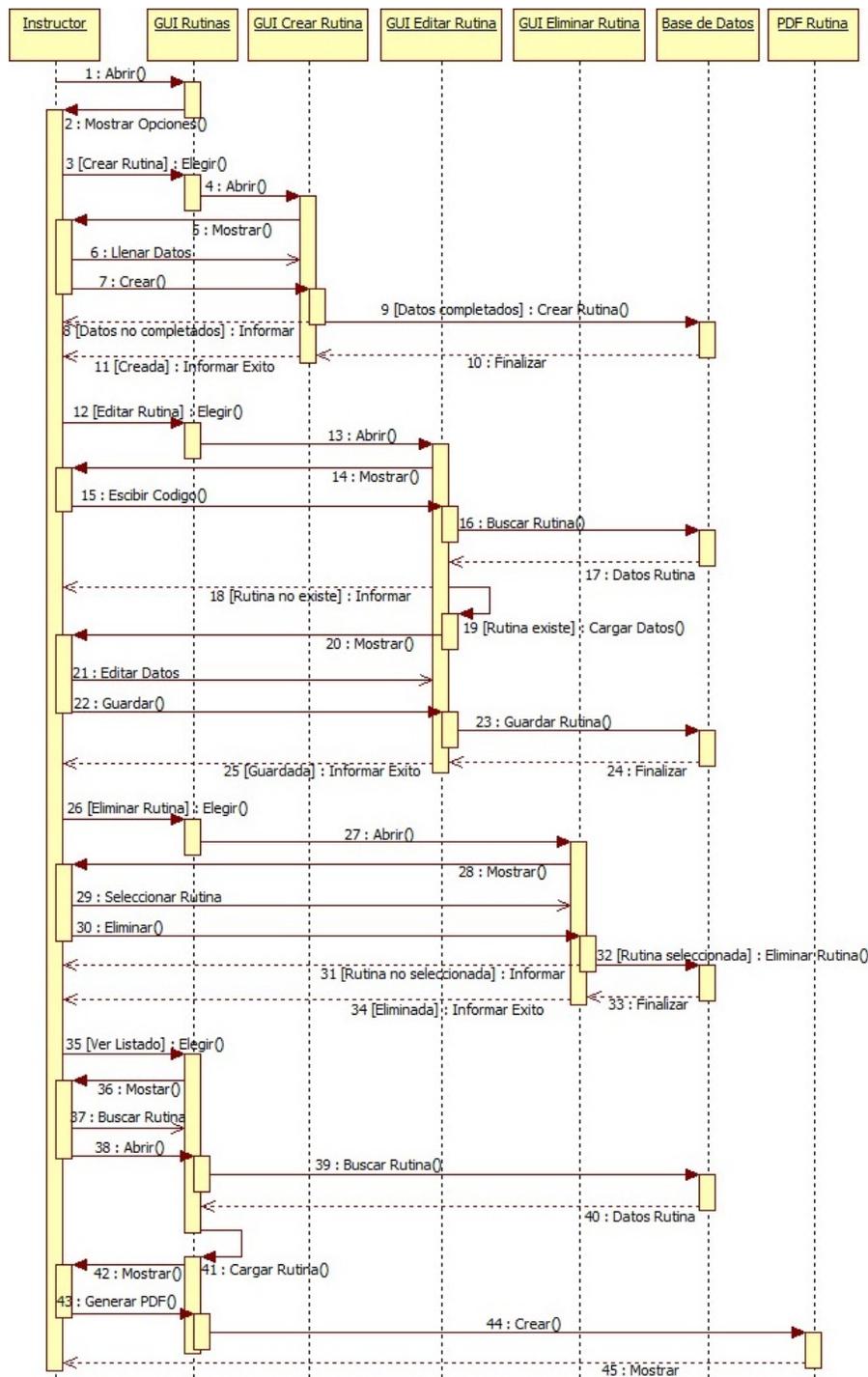


Figura 2.9: Diagrama Secuencias para Administrar Rutina

2.2.8 Administrar Cliente. El caso de uso Administrar Cliente perteneciente al actor Recepcionista muestra la interacción entre el Recepcionista, el Cliente, las

interfaces de clientes, la base de datos y el lector NFC. La Figura 2.10 presenta el diagrama de secuencias del caso de uso Administrar Cliente.

El caso de uso inicia cuando el Recepcionista del gimnasio abre la GUI de Clientes, esta muestra al Recepcionista las opciones que puede realizar. Si el Recepcionista elige Crear Cliente, GUI de Clientes abre la GUI de Crear Cliente y esta es mostrada al Recepcionista. El Recepcionista llena los datos del nuevo cliente y oprime el botón Crear, si algunos datos no fueron completados en los campos, GUI de Crear Cliente informa al Recepcionista del error; si todos los datos de los campos fueron completados, GUI de Crear Cliente crea el cliente en la Base de Datos, cierra la ventana e informa al Recepcionista que el cliente fue creado.

Si el Recepcionista elige Editar Cliente, GUI de Clientes abre la GUI de Editar Cliente y esta es mostrada al Recepcionista. El Recepcionista escribe el número de DNI del cliente y GUI de Editar Cliente busca el cliente en la Base de Datos, si el cliente no existe en la Base de Datos, GUI de Editar Cliente informa al Recepcionista; si el cliente si existe, GUI de Editar Cliente carga los datos del cliente y muestra al Recepcionista. El Recepcionista edita los datos y oprime el botón Guardar, GUI de Editar Cliente guarda los cambios del cliente en la Base de Datos, cierra la ventana e informa al Recepcionista que el cliente fue editado.

Si el Recepcionista elige Eliminar Cliente, GUI de Clientes abre la GUI de Eliminar Cliente y esta es mostrada al Recepcionista. El Recepcionista escribe el número de DNI del cliente y GUI de Eliminar Cliente busca el cliente en la Base de Datos, si el cliente no existe en la Base de Datos, GUI de Eliminar Cliente informa al Recepcionista; si el cliente si existe, GUI de Eliminar Cliente carga los datos del cliente y muestra al Recepcionista. El Recepcionista oprime el botón Eliminar, GUI de Eliminar Cliente borra el cliente en la Base de Datos, cierra la ventana e informa al Recepcionista que el cliente fue eliminado.

Si el Recepcionista elige Buscar Cliente, GUI de Clientes pide al Recepcionista el número de DNI del cliente. El Recepcionista pide el número de DNI al Cliente, escribe el número en la GUI y oprime el botón Buscar, GUI de Clientes abre la GUI de Perfil del Cliente y esta busca al cliente en la Base de Datos, luego carga la información del cliente en el perfil y muestra la ventana al Recepcionista. El Recepcionista verifica el pago en la GUI de Perfil del Cliente, si no hay ningún pago de activación, el Recepcionista informa al Cliente que no puede asignarle una etiqueta NFC; si hay pagos de activación, el Recepcionista entrega al Cliente una etiqueta NFC, oprime el botón Agregar ID en la GUI de Perfil del Cliente y pide al Cliente que pase la etiqueta por el Lector NFC. El Cliente pasa la etiqueta por el Lector NFC, el Lector obtiene la ID de la etiqueta y la envía a la GUI de Perfil del Cliente. la GUI de Perfil del Cliente asigna la ID recibida en la Base de Datos al Cliente del perfil e informa al Recepcionista que la etiqueta fue asignada.

2.2. DIAGRAMAS DE SECUENCIAS

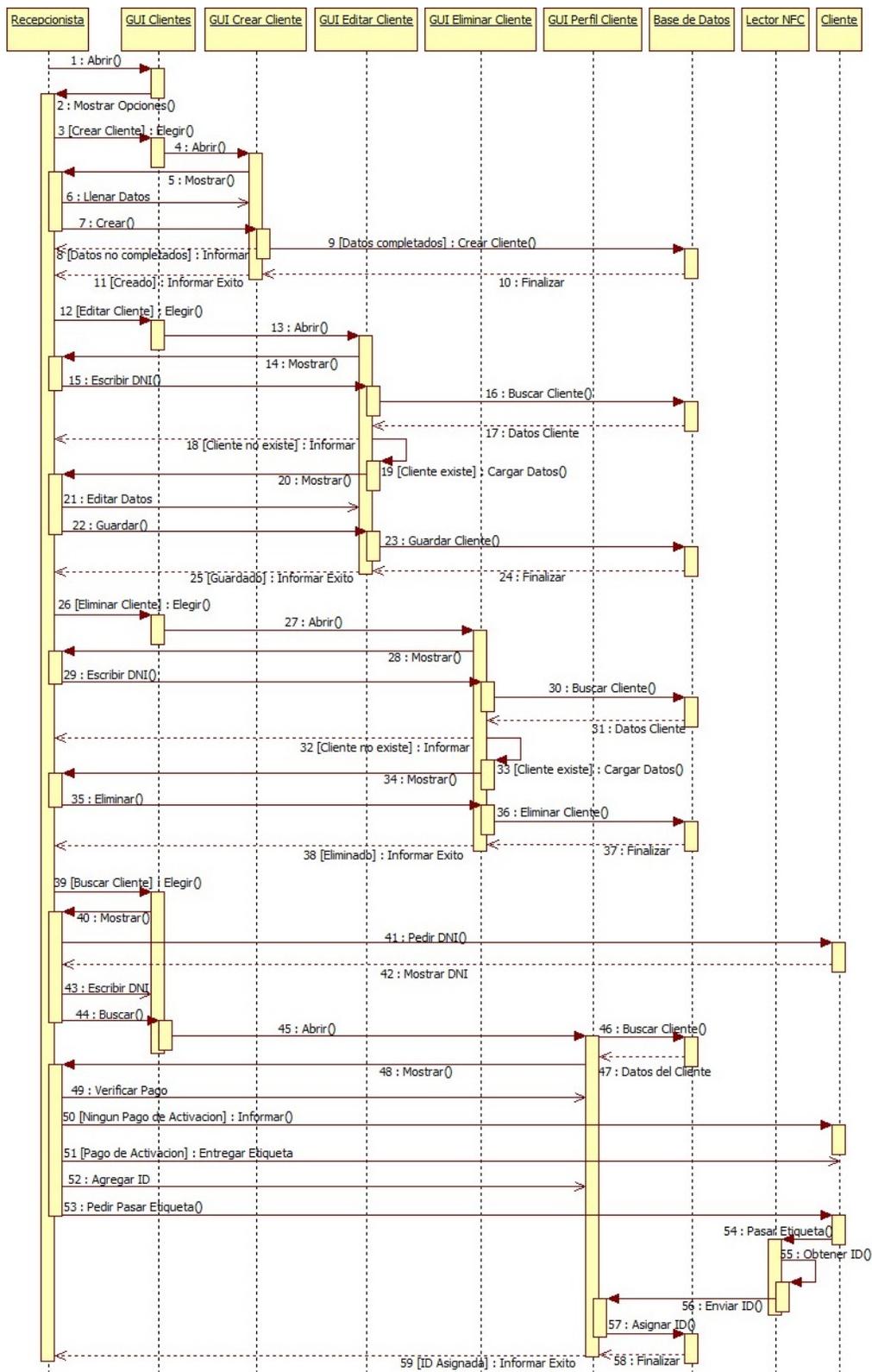


Figura 2.10: Diagrama Secuencias para Administrar Cliente

2.2.9 Administrar Pago. El caso de uso Administrar Pago perteneciente al actor Recepcionista muestra la interacción entre el Recepcionista, las interfaces de pagos y la base de datos. La Figura 2.11 presenta el diagrama de secuencias del caso de uso Administrar Pago.

El caso de uso inicia cuando el Recepcionista del gimnasio abre la GUI de Pagos, esta muestra al Recepcionista las opciones que puede realizar.

Si el Recepcionista elige Crear Pago, GUI de Pagos abre la GUI de Crear Pago y esta es mostrada al Recepcionista. El Recepcionista escribe el número de DNI del cliente y GUI de Crear Pago busca el cliente en la Base de Datos, si el cliente no existe en la Base de Datos, GUI de Crear Pago informa al Recepcionista; si el cliente si existe, GUI de Crear Pago carga el nombre del cliente y muestra al Recepcionista. El Recepcionista llena los datos del nuevo pago y oprime el botón Crear, si algunos datos no fueron completados en los campos, GUI de Crear Pago informa al Recepcionista del error; si todos los datos de los campos fueron completados, GUI de Crear Pago crea el pago del cliente en la Base de Datos, cierra la ventana e informa al Recepcionista que el pago fue creado.

Si el Recepcionista elige Eliminar Pago, GUI de Pagos abre la GUI de Eliminar Pago y esta es mostrada al Recepcionista. El Recepcionista escribe el número de DNI del cliente y GUI de Eliminar Pago busca el cliente en la Base de Datos, si el cliente no existe en la Base de Datos, GUI de Eliminar Pago informa al Recepcionista; si el cliente si existe, GUI de Eliminar Pago carga el nombre del cliente, los pagos del cliente y muestra al Recepcionista. El Recepcionista selecciona el pago y oprime el botón Eliminar, si el pago no fue seleccionado, GUI de Eliminar Pago informa al Recepcionista del error; si el pago fue seleccionado, GUI de Eliminar Pago elimina el pago del cliente en la Base de Datos, cierra la ventana e informa al Recepcionista que el pago fue eliminado.

Si el Recepcionista elige Pagar Saldo, GUI de Pagos abre la GUI de Pagar Saldo y esta es mostrada al Recepcionista. El Recepcionista escribe el número de DNI del cliente y GUI de Pagar Saldo busca el cliente en la Base de Datos, si el cliente no existe en la Base de Datos, GUI de Pagar Saldo informa al Recepcionista; si el cliente si existe, GUI de Pagar Saldo carga el nombre del cliente, los pagos con saldos del cliente y muestra al Recepcionista. El Recepcionista selecciona el pago, escribe la cantidad a saldar y oprime el botón Pagar, si la cantidad no fue escrita, GUI de Pagar Saldo informa al Recepcionista del error; si la cantidad fue escrita, GUI de Pagar Saldo actualiza el pago del cliente en la Base de Datos, cierra la ventana e informa al Recepcionista que el pago fue saldado.

2.2. DIAGRAMAS DE SECUENCIAS

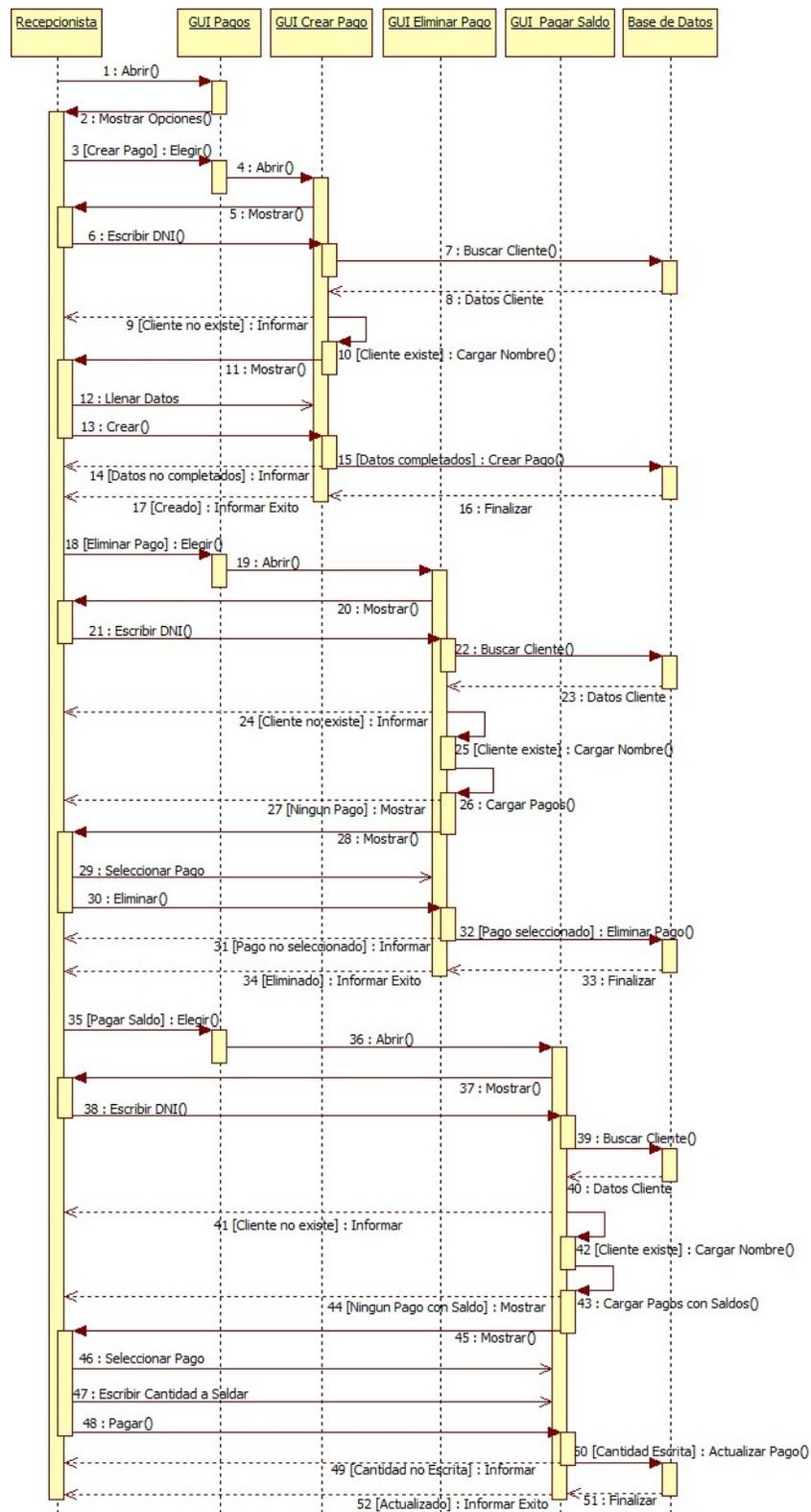


Figura 2.11: Diagrama Secuencias para Administrar Pago

2.3. LECTOR DE ETIQUETAS NFC

El lector de etiquetas NFC está compuesto por 5 componentes de hardware, uno es un Procesador (Arduino UNO R3) y los otros 4 son dispositivos. La Figura 2.12 muestra la estructura física del Lector de etiquetas NFC.

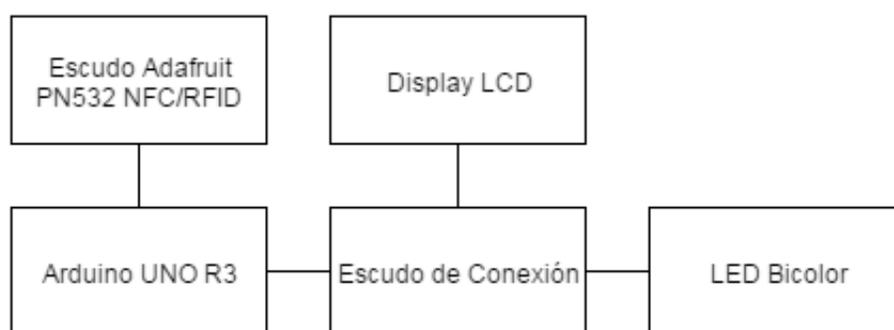


Figura 2.12: Estructura física del Lector de etiquetas NFC

2.3.1 Arduino UNO R3. Es una placa electrónica basada en el microcontrolador ATmega328P, para su funcionamiento se requiere un programa creado en el entorno de desarrollo de Arduino. El programa debe incluir el correcto funcionamiento de todos los dispositivos adheridos al Arduino.

2.3.2 Escudo Adafruit PN532 NFC/RFID. Permite la lectura de etiquetas NFC, detectando la etiqueta que se encuentre próxima al dispositivo, adquiriendo su ID y enviándolo al procesador Arduino UNO R3.

2.3.3 Pantalla LCD. Muestra información de relevancia para el cliente en la pantalla una vez una etiqueta sea leída.

2.3.4 LED bicolor. Permite la visualización de los estados del Lector de Etiquetas NFC.

2.3.5 Escudo de conexión. Es una interfaz de comunicación del Arduino UNO con los dispositivos Pantalla LCD y el LED Bicolor. Este dispositivo se diseña como un circuito impreso en forma de escudo para adherir al Arduino UNO.

2.3.6 Funcionamiento. Una vez que el Lector de etiquetas NFC se conecta al ordenador, el Lector leerá constantemente el puerto serial y se encenderá la pantalla LCD. GymID podrá conectarse al Lector cuando se dé la orden desde la aplicación para escritorio, esta enviará un código de conexión que el Lector reconocerá para activarlo, la Figura 2.13 presenta el diagrama de estados del sistema Lector de Etiquetas NFC.

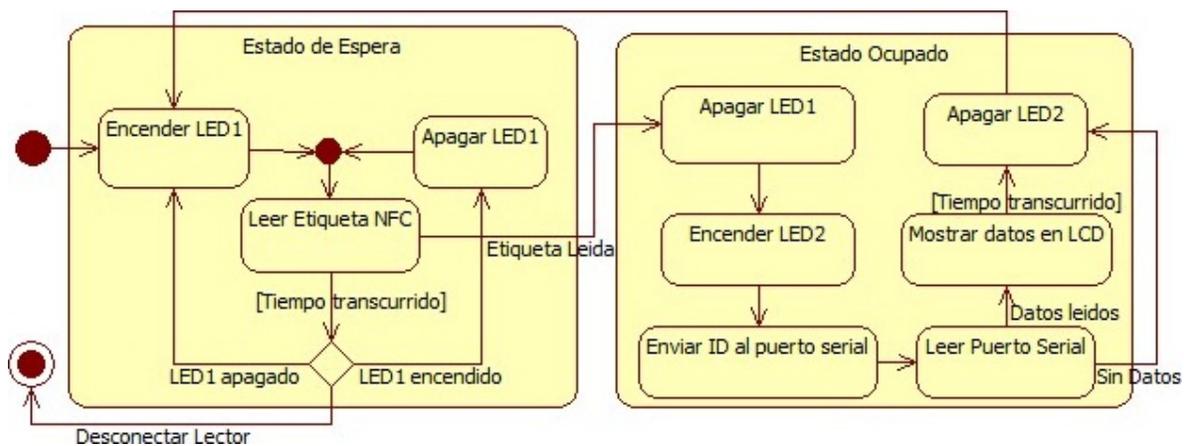


Figura 2.13: Diagrama de Estados del Lector de Etiquetas NFC

El Lector permanecerá bajo 2 estados mientras se encuentre activo:

- **Estado de espera:** La lectura de etiquetas NFC se encuentra activa, el LED amarillo alterna su encendido.
- **Estado ocupado:** La lectura de etiquetas NFC se encuentra desactivada, el LED rojo permanece encendido.

2.3.6.1 Flujo básico. Cuando se activa la lectura de etiquetas, el Lector permanece en el Estado de Espera. Si una etiqueta NFC se coloca sobre el lector, iniciará el Estado Ocupado, este adquirirá el ID de la etiqueta y la enviará al puerto serial, luego de un corto tiempo leerá el puerto serial por una respuesta del ordenador; Si no hay respuesta del ordenador, el Estado Ocupado finalizará e iniciará el Estado de Espera; Si hay una respuesta del ordenador, el Lector mostrará los datos del cliente en la pantalla LCD durante un tiempo, luego finalizará el Estado Ocupado e iniciará el Estado de Espera.

2.3.6.2 Flujo alterno. En cualquier momento el usuario de la aplicación podrá agregar etiquetas NFC a los clientes, al momento de hacerlo el Lector no pertenecerá en ninguno de los estados vistos anteriormente. Cuando el usuario dé la orden en la aplicación, el Lector adquirirá la ID de la primera etiqueta que se aproxime y la enviará al puerto serial, luego iniciará el Estado de Espera.

2.3.7 Programación. Una vez analizado el funcionamiento planteado, se empiezan a escribir las líneas de código en el entorno de desarrollo (IDE) de Arduino.

2.3.7.1 Declaración de variables. Para empezar se escriben las librerías que se utilizarán durante del programa, sean requeridas por el programador o por otras librerías. Se agrega la librería `Adafruit_NFCShield_I2C` que proporciona funcionalidades del escudo `Adafruit PN532 NFC/RFID` bajo el protocolo de comunicación I2C, además esta librería requiere de otra llamada `Wire` que permite la comunicación I2C con los dispositivos de Arduino. Luego se incluye la librería `LiquidCrystal` que permite al Arduino controlar la pantalla LCD.

Se define la variable del lector `nfc` bajo la clase de la librería `Adafruit_NFCShield_I2C` que permitirá la utilización de las funciones de la librería, definiendo la interrupción `IRQ` en el pin 2 y el `reset` en el 3. Al definir el protocolo de comunicación I2C se debe tener en cuenta que la línea de transmisión de datos `SDA` y de reloj `SCL` se establecen por defecto los pines A4 y A5 respectivamente.

Se define la variable `lcd` bajo la clase `LiquidCrystal` que permitirá la utilización de las funciones de la librería. Además se establece el `rs` en el pin 6, el `enable` en el pin 7 y los bits de datos en los pines 10, 11, 12, 13; dejando al pin 13 como el bit más significativo. Este tipo de definición permite al LCD trabajar bajo el modo de 4bits, ignorando los 4 bits menos significativos del LCD.

Se definen las variables que se utilizan durante del programa: `k` es una variable contadora que permitirá borrar la pantalla del LCD cada cierto tiempo; `Led` es una variable booleana que identificará si el led amarillo se encuentra encendido; `Conexión` es la variable que identificará si `GymID` se encuentra conectado al Lector; `Lectura` es la variable que definirá si el Lector ha detectado una etiqueta NFC; `j` es una variable que ayudará en la visualización de los nombres en la pantalla LCD; Por último `c` es una variable que identificará el tipo de dato que es enviado desde el ordenador para ser mostrado en la pantalla LCD.

Se define un símbolo para el mensaje de bienvenida, que se mostrará en la pantalla LCD cuando un cliente se identifique, este representa 2 pesas para hacer referencia al gimnasio.

Las líneas de código de la explicación anterior pueden ser encontradas en la sección B.1 del Anexo B Código de Arduino.

2.3.7.2 Preparación `setup()`. Arduino inicializa el puerto serial definiendo la velocidad de transmisión serial a 115200, También se inicializa el dispositivo de lectura NFC y la pantalla LCD de 20x4.

Luego se realizan unas configuraciones iniciales para los dispositivos conectados: se configura el lector para que pueda leer etiquetas NFC; se crea el símbolo del mensaje de bienvenida en la memoria del LCD; por último se establecen los pines 8 y 9 del Arduino como pines de salida de datos para controlar el LED (Amarillo y Rojo respectivamente).

La líneas de código de la explicación anterior pueden ser encontradas en la sección B.2 del Anexo B Código de Arduino.

2.3.7.3 Bucle principal loop(). El bucle principal inicia verificando la conexión con GymID; si no hay comunicación, Arduino leerá el puerto serial hasta que GymID establezca la comunicación por medio de un código de conexión establecido como '\$@%'.

Cada vez que la función de bucle principal se repita, el LED alternará su encendido. Además cuando el bucle se repite 8 veces una vez que la información del cliente ha sido mostrada en la pantalla LCD, la pantalla se borra (24 segundos después de haber terminado el estado ocupado).

Una vez por ciclo se restablecen las variables de la etiqueta y el Arduino activa la lectura de etiquetas NFC por 3 segundos; si una etiqueta fue detectada iniciará el estado ocupado. El número de identificación de la etiqueta se guarda en la variable ID como una cadena de texto y luego se envía al puerto serial. El lector no diferencia si una etiqueta es para agregarla a un cliente o para registrar una visita al momento de leer la ID, por lo tanto siempre que se realiza una lectura y esta es enviada al ordenador, el lector espera la respuesta en el puerto serial. Si el dato es '>', es porque el lector adquirió una ID para agregarla a un cliente y la operación finalizó con éxito, a diferencia del dato '<', ya que le indica que la operación de agregar la ID falló. Por último si el primer dato que se escribió en el puerto serial no es ninguno de los dos, el lector reconoce que un cliente se ha identificado y muestra el mensaje de bienvenida en la pantalla LCD con la función Saludo(), además se restablecen las variables de información del cliente.

Inicia ciclo de lectura de datos, se realiza una verificación de los datos leídos en el puerto serial, en este punto se clasifican los datos por medio de una decodificación de la escritura hecha por la aplicación. Cada '&' diferencia el tipo de datos leído y luego prepara los datos a escribir dependiendo de la activación de la cuenta del cliente. Cada tipo de datos es escrito en la ubicación correspondiente de la pantalla LCD. Si el nombre del cliente sobrepasa los 20 caracteres, se realiza un efecto de desplazamiento automático del texto en el LCD, por cada letra se tarda 500 milisegundos adicionales. Cuando la última letra haya sido escrita, la pantalla LCD muestra por 1 segundo el final del nombre, luego mostrará los primeros 20 caracteres del nombre. Mientras GymID continúe conectado, finalizará el estado ocupado e iniciará el estado de espera.

Se verifica si hay datos en el puerto serial; si hay datos y los datos son el código de desconexión '%@\$', el Lector se desconectará de GymID. En este punto el bucle

principal finaliza y vuelve a empezar el ciclo.

Las líneas de código de la explicación anterior pueden ser encontradas en la sección B.3 del Anexo B Código de Arduino.

2.3.7.4 Función de bienvenida Saludo(). El Saludo() es la función que se llama al momento que GymID envíe los datos del cliente por el puerto serial, este muestra un mensaje de bienvenida al cliente que ingresa. Arduino escribirá un mensaje de bienvenida al cliente en la pantalla LCD con una animación utilizando el símbolo de las pesas, esta animación dura 3 segundos y hace parte del estado ocupado del dispositivo.

Las líneas de código de la explicación anterior pueden ser encontradas en la sección B.4 del Anexo B Código de Arduino.

2.3.8 Escudo de conexión. Teniendo en cuenta los pines de Arduino a utilizar por los dispositivos Pantalla LCD y LED Bicolor, se realiza el diseño del escudo de Conexión en el programa PCB Wizard.

Primero se hace el diagrama esquemático del circuito teniendo como referencia la placa del Arduino UNO y los pines a utilizar en la programación. Se agregan los terminales disponibles para los conectores, en este caso se utiliza un conector de 3 pines para el LED Bicolor y 2 conectores de 8 pines cada uno para la pantalla LCD. La Figura 2.14 presenta el diagrama esquemático del Escudo de Conexión.

Además el circuito utiliza 3 resistencias de protección requeridas por cada ánodo del LED y una para la iluminación de fondo de la pantalla LCD. Luego 2 resistencias adicionales hacen un divisor de tensión para configurar el contraste del LCD anteriormente calculado con un potenciómetro. El elegir los caminos menos complejos causa que el orden de pines de los dispositivos a conectar no coincidan directamente, es importante tener esto en cuenta al momento de realizar el cableado con los dispositivos. La Figura 2.15 presenta el diseño PCB definitivo del escudo de conexión.

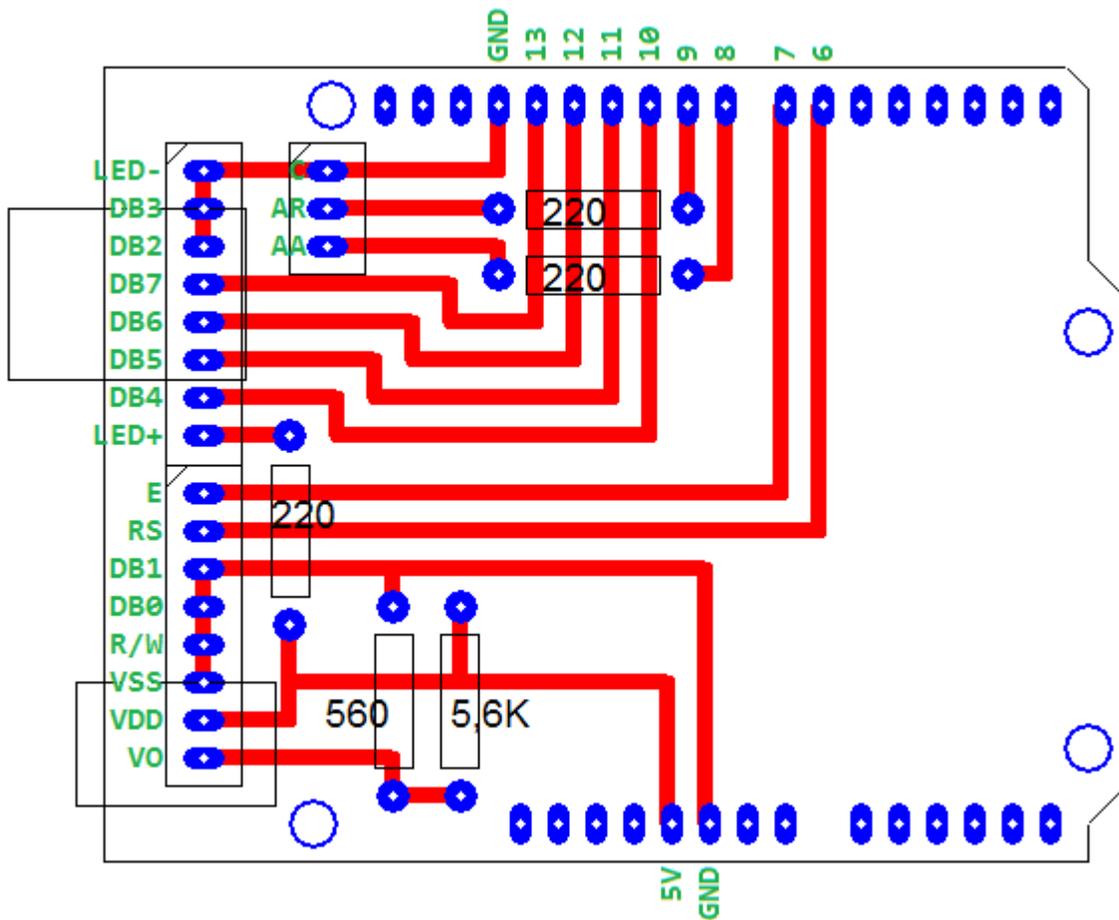


Figura 2.14: Diseño esquemático del Escudo de Conexión

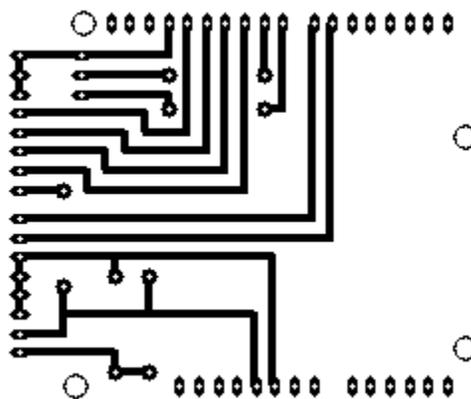


Figura 2.15: Diseño PCB del Escudo de Conexión

3. IMPLEMENTACIÓN

3.1. DIAGRAMA DE DESPLIEGUE

Un diagrama de despliegue modela la disposición física de los artefactos de software en nodos (hardware). Existe 2 tipos de nodos, uno es de tipo Procesador y permite ejecutar un componente de software, el otro es de tipo Dispositivo y no ejecuta componentes. La Figura 3.1 muestra el diagrama de despliegue del sistema.

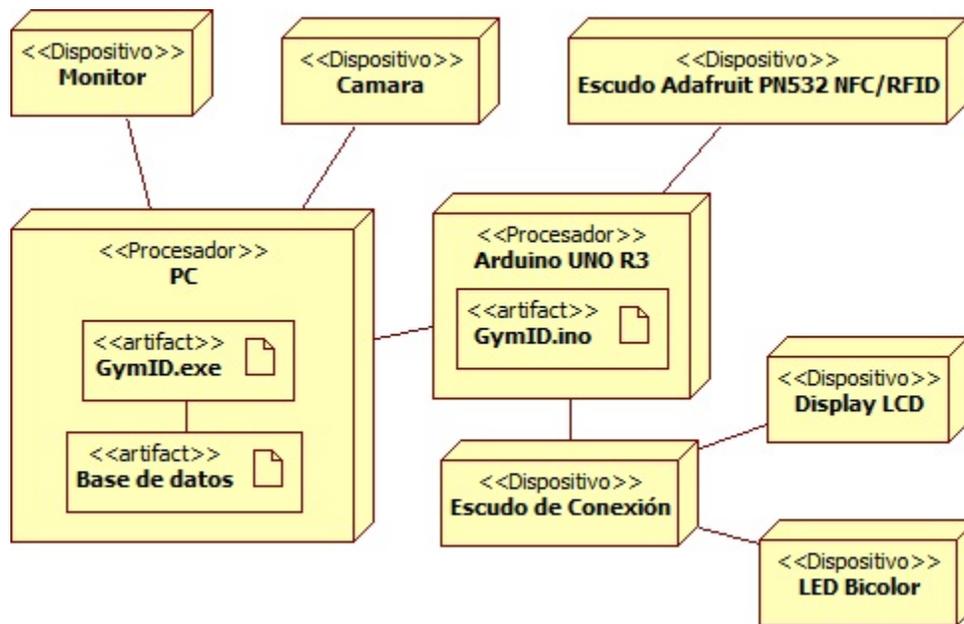


Figura 3.1: Diagrama de Despliegue del Sistema

El nodo PC que puede tratarse de un ordenador de escritorio o portátil que integra un Monitor y una Cámara Web para el funcionamiento del componente de software `GymID.exe`, además está conectado por medio de un cable USB tipo AM/BM al Dispositivo Lector de Etiquetas NFC, conformado por el Arduino UNO R3 donde se encuentra el componente `GymID.ino` y este se conecta directamente con el escudo Adafruit PN532 NFC/RFID y el escudo de Conexión, este último provee una interfaz de comunicación entre los dispositivos Display LCD y LED Bicolor con el Arduino UNO R3.

3.2. DIAGRAMA DE COMPONENTES

El diagrama de componentes muestra cómo un sistema de software es dividido en componentes y sus dependencias. Los componentes tienen interfaces que son operaciones realizadas por un componente y utilizadas por otros. La Figura 3.2 presenta el diagrama de componentes del sistema. El sistema está dividido en tres componentes principales: GymID.exe, GymID.ino y la base de datos de MySQL.

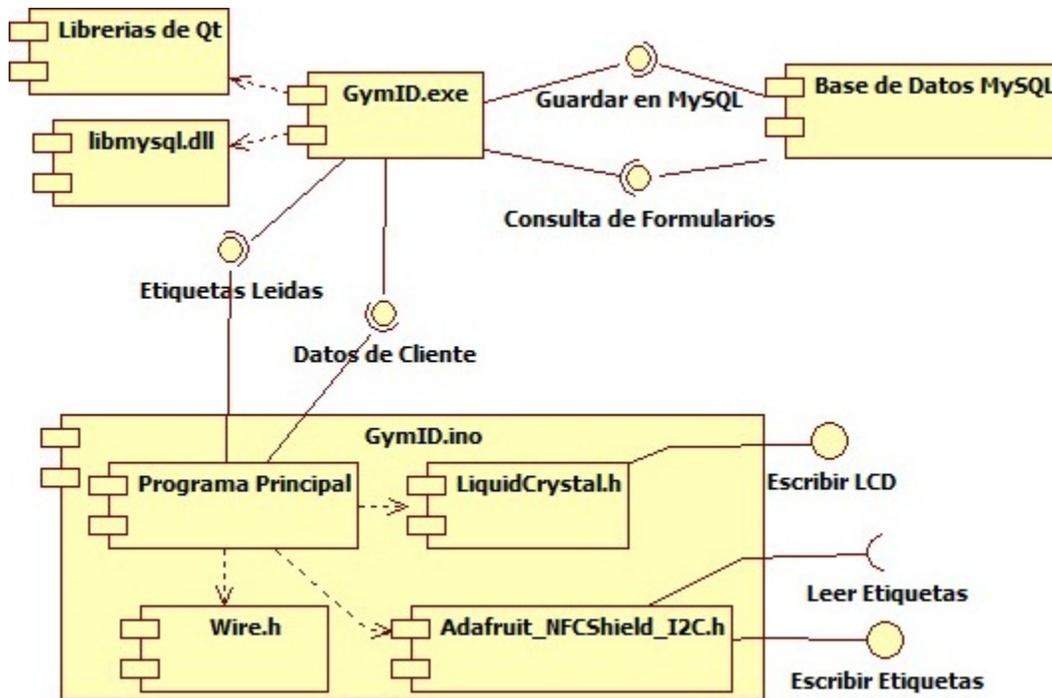


Figura 3.2: Diagrama de Componentes del Sistema

GymID.exe es la aplicación para escritorio que depende de las Librerías de Qt y libmysql.dll. Las Librerías de Qt son todas aquellas librerías de Qt Creator para que la aplicación pueda funcionar en el sistema operativo: multimedia, formatos de imagen, audio, soporte de impresión, drivers de SQL, de puerto serial, de red, Qt5Core, Qt5Gui, Qt5Widget, etc. Mientras que libmysql.dll es la librería que conecta la aplicación al servidor de MySQL. Estas librerías son indispensables para el funcionamiento de la aplicación.

GymID.ino es la aplicación para el Arduino, que esta formada por un Programa Principal de código fuente que depende de LiquidCrystal.h, Wire.h y Adafruit NFCShield I2C.h. Todas son librerías dentro del programa, LiquidCrystal.h facilita la escritura en el display LCD, Wire.h habilita la comunicación I2C entre dispositivos de Arduino, esto

último para comunicarse con el escudo lector de etiquetas NFC, Mientras que Adafruit NFCShield I2C.h controla el funcionamiento del lector NFC.

Por último la Base de Datos de MySQL que está integrada por tablas y registros de bases de datos, solo se expresa como un solo componente para simplificar el diagrama ya que su funcionamiento es sencillo y no es relevante para el análisis de componentes del sistema.

Luego de haber presentado el funcionamiento en software del sistema, a continuación se presentara un análisis detallado de la estructura interna del componente GymID.exe, La Figura 3.3 muestra el diagrama de componentes de la aplicación GymID.exe.

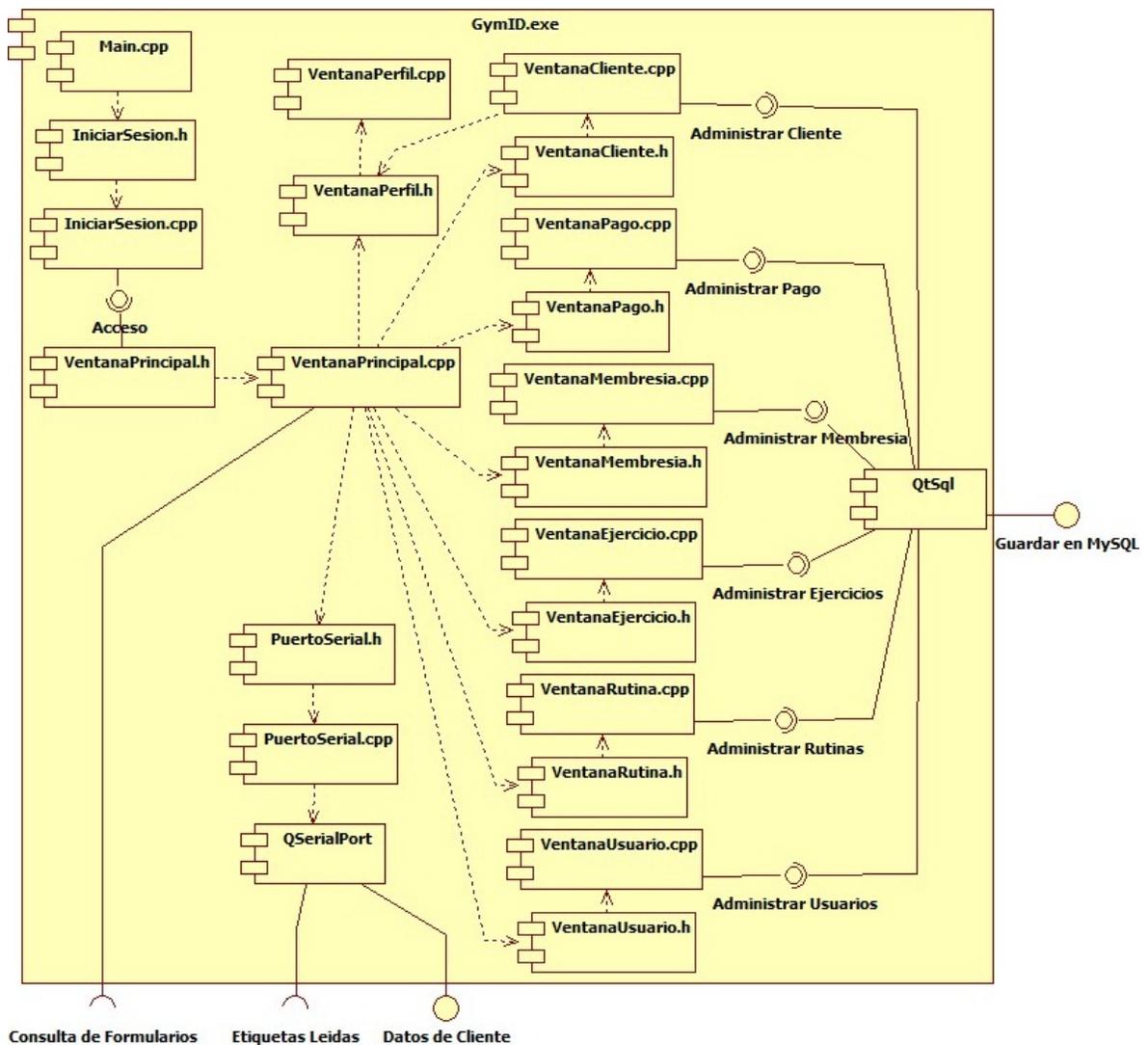


Figura 3.3: Diagrama de Componentes de GymID.exe

Cada componente hace referencia al código fuente del programa, los encabezados (.h)

3.3. BASE DE DATOS

y la implementación (.cpp) son la representación de las clases más importantes creadas para la aplicación, a excepción de Main.cpp que es el fichero de inicio del programa.

El diagrama muestra las dependencias que tiene cada fichero. Main.cpp inicia el programa llamando a la clase IniciarSesion, esta contiene un algoritmo que permite el acceso a la VentanaPrincipal del programa cuando el usuario ha escrito correctamente el nombre de usuario y contraseña. Para esto IniciarSesion se conecta a la base de datos para verificar los datos del usuario, esta y otras operaciones fueron omitidas en el diagrama para simplificar el modelo.

VentanaPrincipal muestra el menú de acciones de aplicación, esta utiliza la clase PuertoSerial para activar la comunicación (lectura y escritura) del puerto serial por medio de la librería QSerialPort. También se realizan consultas de formularios de la base de datos para ser mostradas en tablas de VentanaPrincipal, los perfiles de los clientes pueden ser visualizados desde estas tablas por medio de la clase VentanaPerfil.

Igualmente desde VentanaPrincipal se pueden abrir cada una de las ventanas de administración para agregar, editar o eliminar datos de la base de datos por medio de las clases VentanaCliente, VentanaPago, VentanaMembresia, VentanaEjercicio, VentanaRutina, VentanaUsuario. Cada ventana utiliza la librería de QSql que permite la ejecución de instrucciones a la base de datos desde la aplicación. Al ser VentanaCliente la encargada de manejar toda la información de los clientes, también utiliza la clase VentanaPerfil.

3.3. BASE DE DATOS

El modelo de la base de datos de MySQL está compuesto por un modelo de entidad-relación que incluye Vistas, Rutinas y Disparadores (Triggers) de las tablas. La Figura 3.4 muestra el modelo de la base de datos del sistema.

El diagrama muestra un modelo de Entidad-Relación avanzado o como MySQL lo denomina "mejorado", en la cual las entidades están representados por tablas que contienen atributos. Algunos atributos son llaves primarias (símbolo de llave) y llaves foráneas (símbolo rojo).

En el caso de la entidad Entrenamiento que contiene 2 llaves primarias en color rojo, significa que son llaves primarias para la entidad pero fueron creadas a partir de llaves externas (foráneas), y se evidencia con la continuidad de la línea en la relación de esa entidad. Esto ocurre porque la entidad Entrenamiento es el resultado entre la relación de Rutina y Ejercicio que es una relación de muchos a muchos.

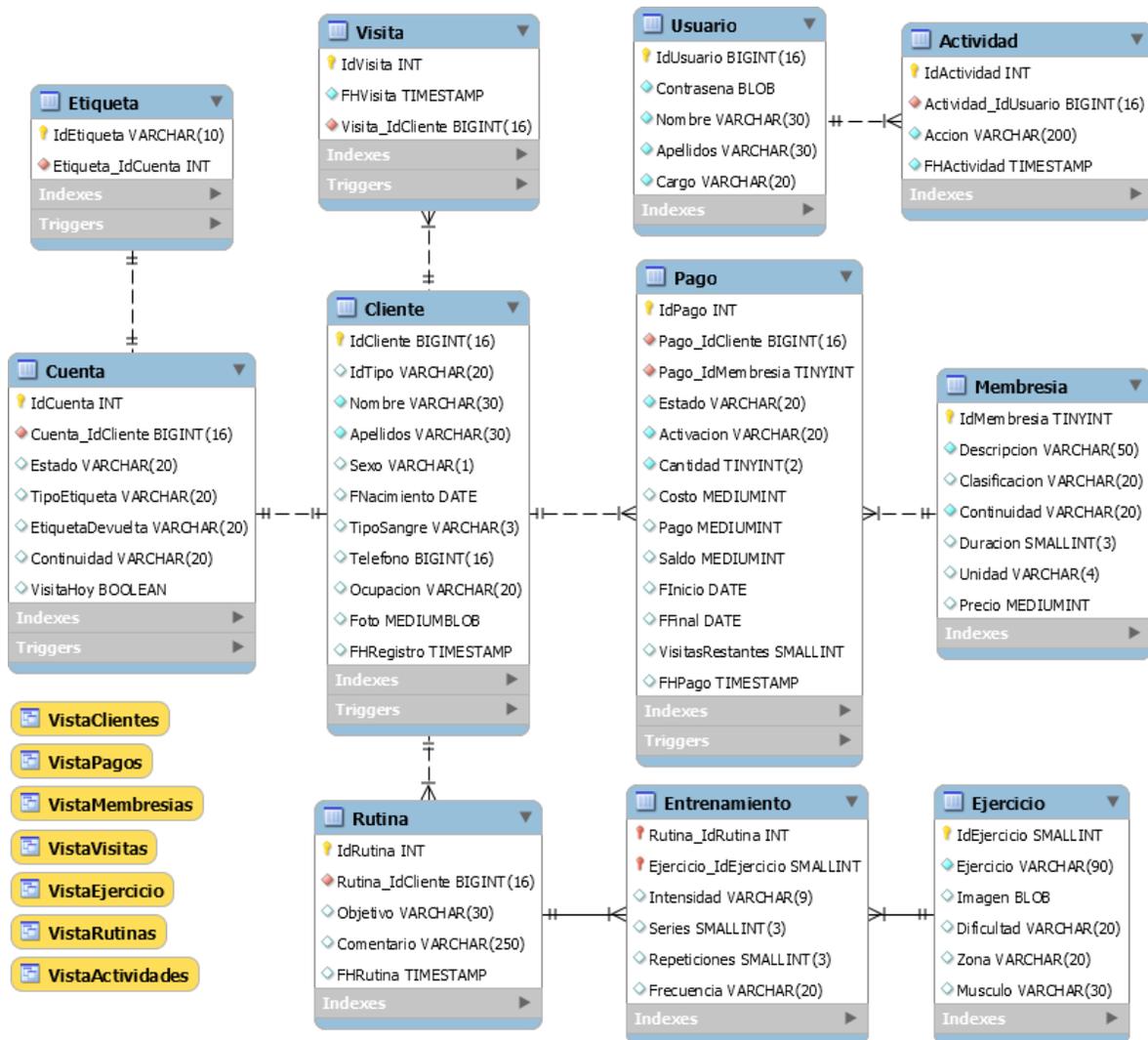


Figura 3.4: Modelo relacional del banco de datos

La cardinalidad en el diagrama se representa por símbolos en la relación, en caso de 1 se representa con dos barras paralelas y N (muchos) se representa con tres líneas que se dividen de la relación. Por Ejemplo en la relación Cliente y Pago significa que 1 cliente puede tener muchos pagos asignados.

Además MySQL diferencia entre los valores de los atributos que pueden ser nulos y los que no, con el símbolo rombo de color blanco y azul claro respectivamente.

Cada atributo se identifica con un tipo de dato que puede ser enteros como INT, BIGINT, SMALLINT, TINYINT, etc. Cada uno se diferencia por el tamaño en Byte que utiliza. El diagrama también incluye datos de tipo VARCHAR que admite strings (arreglo de caracteres), DATE que admite fechas, TIMESTAMP que admite tiempos y

fechas, BOOLEAN dato booleano, además variables de tipo BLOB que admiten objetos binarios de gran tamaño, esto se utiliza para guardar las fotografías y guardar datos cifrados como las contraseñas de los usuarios.

3.3.1 Tablas. Las tablas almacenan la información de la base de datos organizadamente, cada tabla se distingue de las otras por su nombre y almacenan información por medio de atributos que la identifican. A continuación se presenta cada tabla con sus atributos correspondientes:

3.3.1.1 Cliente. Esta tabla almacena la información del cliente, tiene una llave primaria llamada IdCliente que es en otras palabras el documento de identidad de la persona, además las clasifica por medio de IdTipo (tipo de documento). Aquí se puede encontrar los datos personales Nombre, Apellidos, Sexo, FNacimiento (fecha de nacimiento), TipoSangre (tipo de sangre), Teléfono, Ocupación. Igualmente almacenara la fotografía del cliente por medio de Foto y la fecha de registro del cliente por medio de FHRegistro.

3.3.1.2 Pago. Esta tabla almacena la información de los pagos realizado por los clientes, su llave primaria es IdPago que es un código entero auto incremental. También tiene dos llaves foráneas, procedente de Cliente (Pago_IdCliente) y Membresía (Pago_IdMembresia). El resto de información son características del pago como Estado, Activación (tipo de activación, que puede ser: Inmediata, Postergada o Eventual), Cantidad (numero de veces que se paga la membresía), Costo (precio del pago), Pago (monto que paga), Saldo (monto que queda debiendo), FInicio (fecha que inicia la membresía que paga), FFinal (fecha que finaliza la membresía que paga), VisitasRestantes (cantidad de días que faltan hasta que acabe el pago) y FHPago (fecha de creación del pago).

3.3.1.3 Membresia. Esta tabla almacena la información de las membresías (tipos de pago que permite el gimnasio), la llave primaria es IdMembresia que es un código especificado por el usuario. Cada membresía se caracteriza por tener una Descripción (nombre de la membresía), Clasificación (puede ser: Estudiante o Particular u Otro), Continuidad (tipo de ingreso, puede ser Normal: durante un periodo de tiempo o Adaptable: se cobra por días asistidos), Duración (cantidad de Unidad), Unidad (puede ser: Día o Mes) y Precio (precio de la membresía).

3.3.1.4 Cuenta. Esta tabla almacena información de la cuenta de los clientes, la llave primaria es IdCuenta que es un código entero auto incremental. La cuenta tiene

una llave foránea de Cliente (`Cuenta_IdCliente`), además almacena datos del Estado, `TipoEtiqueta` (tipo de etiqueta, puede ser: Tarjeta, Llavero o Pulsera), `EtiquetaDevuelta` (Estado de la etiqueta), `Continuidad` (tipo de ingreso de la membresía activa) y `VisitaHoy` que verifica si el cliente ya ingreso el día de “hoy” al gimnasio (solo aplica a cuentas con continuidad Adaptable).

3.3.1.5 Etiqueta. Esta tabla almacena información de las etiquetas entregadas a los clientes, la llave primaria es `IdEtiqueta` que es la ID interna que tiene cada etiqueta y se almacena como una cadena de texto porque son valores hexadecimales. También tiene la llave foránea de Cuenta (`Etiqueta_IdCuenta`).

3.3.1.6 Visita. Esta tabla almacena información de las visitas que realizan los clientes al gimnasio, la llave primaria es `IdVisita` que es un código entero auto incremental. Además tiene una llave foránea de Cliente (`Visita_IdCliente`) y almacena información de `FHVisita` que es la fecha y la hora del día que se realiza la visita.

3.3.1.7 Rutina. Esta tabla almacena información de las rutinas de entrenamiento de los clientes y se enfoca en la información general de la rutina. La llave primaria es `IdRutina` que es un código especificado por el usuario. Además tiene una llave foránea de Cliente (`Rutina_IdCliente`) y atributos como `Objetivo` (meta a alcanzar por la rutina), `Comentario` (indicaciones del instructor) y `FHRutina` que guarda la fecha y hora de creación de la rutina.

3.3.1.8 Entrenamiento. Esta tabla también almacena información de las rutinas de entrenamiento de los clientes, pero se enfoca en el trabajo de entrenamiento. Además cuenta con dos llaves primarias que son externas y vienen de Rutina (`Rutina_IdRutina`) y Ejercicio (`Ejercicio_IdEjercicio`). También tiene como atributos la `Intensidad` (peso o duración del ejercicio), `Series` (cantidad de veces que se realiza una secuencia del mismo ejercicio), `Repeticiones` (cantidad de veces que se realiza un ejercicio) y la `Frecuencia` son los días que se debe hacer el ejercicio (este valor es codificado por el programa para que pueda guardarse como una línea de caracteres).

3.3.1.9 Ejercicio. Esta tabla almacena información de los ejercicios para que se agreguen en las rutinas de entrenamiento. La llave primaria es `IdEjercicio` que es un código especificado por el usuario. Además tiene atributos como `Ejercicio` (nombre del ejercicio), `Imagen` (imagen instructivo del ejercicio), `Dificultad` (puede ser: Inicial, Medio o Avanzado), `Zona` (lugar del cuerpo que ejercita) y `Musculo` (musculo que ejercita).

3.3.1.10 Usuario. Esta tabla almacena la información del usuario del programa, la llave primaria es IdUsuario que es el documento de identidad del usuario. Además se pueden encontrar atributos como Contraseña, Nombre, Apellidos y Cargo que es el puesto que desempeña en el gimnasio (Administrador, Recepcionista o Instructor).

3.3.1.11 Actividad. Esta tabla almacena la información de las actividades realizadas por los usuarios, la llave primaria es IdActividad que es un código entero auto incremental. También tiene la llave foránea de Usuario (Actividad_IdUsuario), el atributo Acción en el que se guarda cada acción realizada por el usuario en el programa y FHActividad en el que se guarda la fecha y la hora del día de la acción.

3.3.2 Vistas. Las vistas son tablas virtuales que permiten organizar fácilmente la información de varias tablas por medio de consultas establecidas. Cada vista es creada con el objetivo de ser llamada por la aplicación de escritorio, mostrando la información en listados de forma presentada para el usuario. Organizando la información relevante e ignorando datos que no son de interés para el usuario.

3.3.3 Rutinas. El modelo utiliza una sola rutina llamada ‘ActualizarBaseDeDatos’, como su nombre lo dice esta rutina cumple la función de actualizar la base de datos con el paso de los días. Cada día la rutina se activa una vez y actualiza el Estado de los pagos, también actualiza las VisitasRestantes de los pagos que tienen continuidad Normal y reinicia la variable VisitaHoy de las cuentas de los clientes con continuidad Adaptable.

3.3.4 Disparadores. Los disparadores son objetos de la base de datos asociado a tablas y son activados cuando un determinado evento ocurre en la tabla. Esto permite que se realicen acciones antes o después de que un evento como crear, actualizar o eliminar datos de las tablas se hagan.

La tabla Cliente dispone de un disparador llamado ‘CrearCuenta_AgregarCliente’, se activa después de que un nuevo cliente haya sido agregado. La función creará una nueva cuenta asignada a ese cliente.

La tabla Pago dispone de 2 disparadores, el primero ‘ActualizarCuenta_ActualizarEstado’ se activa después de que un pago haya sido actualizado. Esta función verifica si el pago cambió su atributo Estado, si el Estado cambio de Disponible a Activo, la función actualiza el Estado y la Continuidad de la Cuenta del cliente; si el Estado cambio de Activo a Vencido, la función verifica si existen pagos disponibles y actualiza

el Estado, la Continuidad y la VisitaHoy la Cuenta del cliente. El segundo ‘ActualizarCuenta_EliminarPago’ se activa después de que un pago haya sido borrado. Si el pago borrado tenía el Estado Activo, la función verifica si existen pagos disponibles y actualiza el Estado y la Continuidad de la Cuenta del cliente; si el pago borrado tenía el Estado Disponible, la función verifica si existe un pago activo o pagos disponibles y actualiza el Estado y la Continuidad de la Cuenta del cliente.

La tabla Visita dispone de un disparador ‘ActualizarVisitasHoy_AgregarVisita’, se activa después de que se agregue una visita a la base de datos. La función verifica en la Cuenta del cliente si el Estado es activo y la Continuidad es adaptable, luego verifica si ya realizó visitas ese día; si no ha realizado ninguna visita, la función actualiza el atributo VisitaHoy de la Cuenta del cliente.

La tabla Cuenta dispone de un disparador ‘ActualizarVisitasR_ActulizarVisitaHoy’, se activa después de que la cuenta haya sido actualizada. Esta función verifica si la Cuenta cambió su atributo VisitaHoy, si VisitaHoy cambio de falso a verdadero, la función actualiza las VisitasRestantes de la tabla Pago, luego verifica el valor de ese atributo; si las VisitasRestantes son igual a cero, la función verifica si el próximo pago también tiene continuidad adaptable. Si existe un pago próximo con continuidad adaptable, la función actualiza la fecha de inicio de ese pago igual a la fecha de día siguiente.

Por último la tabla Etiqueta dispone de un disparador llamado ‘ActualizarCuenta_EliminarEtiqueta’, este se activa después de que una etiqueta haya sido borrada. La función actualizara el TipoEtiqueta y la EtiquetaDevuelta de la tabla Cuenta.

3.4. MONTAJE Y PRUEBAS DEL LECTOR DE ETIQUETAS NFC

Esta sección muestra un registro fotográfico del montaje del dispositivo de lector de etiquetas NFC. Cuando se ha finaliza la creación del escudo de conexión, se procede a montar el dispositivo, la Figura 3.5 muestra el escudo de conexión.

Se ensambla cada pieza del dispositivo y se coloca dentro de la carcasa del lector, la Figura 3.6 muestra el montaje del lector de etiquetas NFC. La carcasa fue creado especialmente para este proyecto por medio de un trabajo en impresión 3D en material PLA a una calidad normal, la empresa realizadora de trabajo se llama Realiza 3D, la cual recibió el diseño luego de haberse creado las medidas adecuadas para el dispositivo en el entorno de desarrollo de AutoCAD, para más información sobre el diseño de la carcasa, revisar el Anexo C Diseño de la cubierta del lector NFC.

3.4. MONTAJE Y PRUEBAS DEL LECTOR DE ETIQUETAS NFC

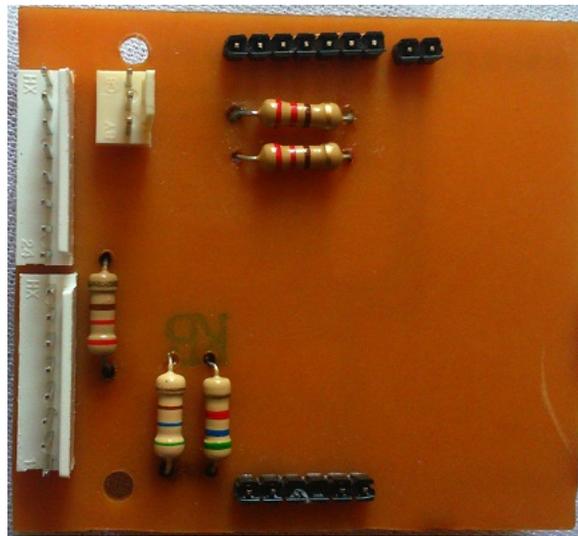


Figura 3.5: Escudo de conexión



Figura 3.6: Montaje del lector de etiquetas NFC

Por último se procede a hacer las pruebas del funcionamiento principal del dispositivo, se configura la aplicación y la base de datos para hacer las pruebas correspondientes,

cabe mencionar que el lector es controlado por las funciones de la aplicación GymID y su funcionamiento depende del evento de captura de ID de etiquetas próximas. Primero se pone el dispositivo en estado de espera, la Figura 3.7a muestra el lector en estado de espera. Luego se realiza la prueba de agregar una etiqueta a un cliente, la Figura 3.7b muestra el lector luego de haber capturado la ID de la etiqueta que se quiere agregar.



(a) Lector en estado de espera

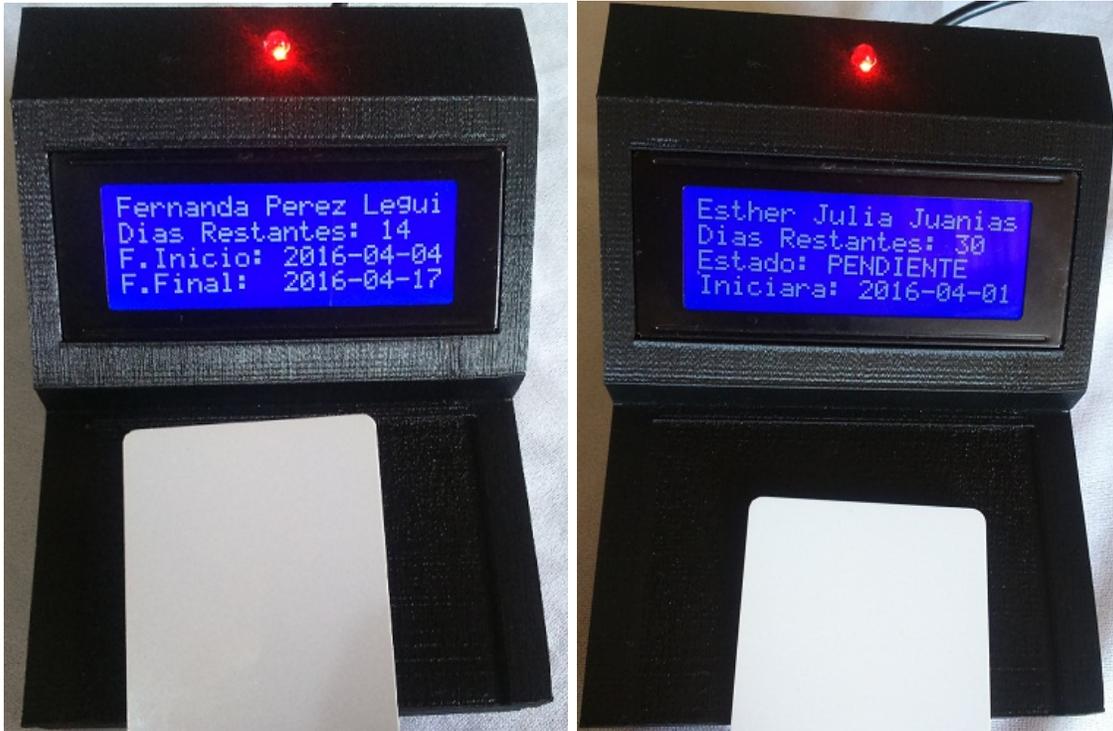
(b) Lector en función de agregar ID a cliente

Figura 3.7: Lector de etiquetas NFC en funcionamiento

Se procede a leer etiquetas de diferentes clientes cuando realizan una visita al gimnasio para mostrar el estado ocupado del lector, la Figura 3.8 muestra el formato de visualización de las posibles lecturas en las visitas.

El lector muestra al cliente la información de su cuenta dependiendo del estado que se encuentre, si la cuenta se encuentra en estado Activo, este muestra el nombre completo del cliente, los días restantes, la fecha de inicio y final del pago activo, como se muestra en la Figura 3.8a. Si la cuenta se encuentra en estado Pendiente, este muestra el nombre completo del cliente, los días restantes, el estado de la cuenta y la fecha que inicia próximo pago, como se muestra en la Figura 3.8b. Por último si la cuenta se encuentra en estado Inactivo, este muestra el nombre completo del cliente, los días restantes, el estado de la cuenta y la fecha que finalizó el último pago, como se muestra en la Figura 3.8c.

3.4. MONTAJE Y PRUEBAS DEL LECTOR DE ETIQUETAS NFC



(a) Cliente en estado Activo

(b) Cliente en estado Pendiente



(c) Cliente en estado Inactivo

Figura 3.8: Formato de posibles lecturas

3.5. APLICACIÓN PARA ESCRITORIO

GymID es la aplicación para escritorio creada a partir de las necesidades del sistema, su diseño está basado en el diagrama de clases y cumple cada uno de los casos de uso planteados. Esta aplicación puede ser operada por 3 tipos usuarios del gimnasio, cada uno tiene acceso a funciones limitadas por permisos dentro del sistema que le permiten administrar la base de datos. La aplicación puede ser instalada en diversos ordenadores para conectarse al servidor de la base de datos por medio de una red de área local.

3.5.1 Entrar al Sistema. Cada usuario de la aplicación debe identificarse para poder acceder a las funciones de GymID, estas funciones cambiarán dependiendo del tipo de usuario que inicie la sesión. Para poder ingresar es necesario tener una cuenta, la cual puede ser creada con la función de Primer Administrador en cualquier momento después de haber instalado la aplicación o a partir de otro usuario Administrador que pueda crearla. La Figura 3.9 muestra la interfaz gráfica para entrar al sistema GymID.



Figura 3.9: Interfaz de Entrar al Sistema

3.5.2 Ventana Principal. La ventana principal es modificada a medida que cambia el tipo de usuario que ingresa. El usuario Administrador tiene acceso a todas las funciones de GymID, a diferencia de los usuarios Recepcionista e Instructor, que sus funciones son limitadas. La Figura 3.10 muestra la interfaz gráfica de la ventana principal para un usuario Administrador.

La ventana está organizada por zonas que permiten una mejor visualización de los componentes que la integran:

La Barra de Menú ubicada en la parte superior está organizada por pestañas (Archivo, Edición, Ver, Administrar y Ayuda) que incluyen todas las funciones existentes de la aplicación GymID. A continuación, la Barra de Herramientas implementada con iconos

3.5. APLICACIÓN PARA ESCRITORIO

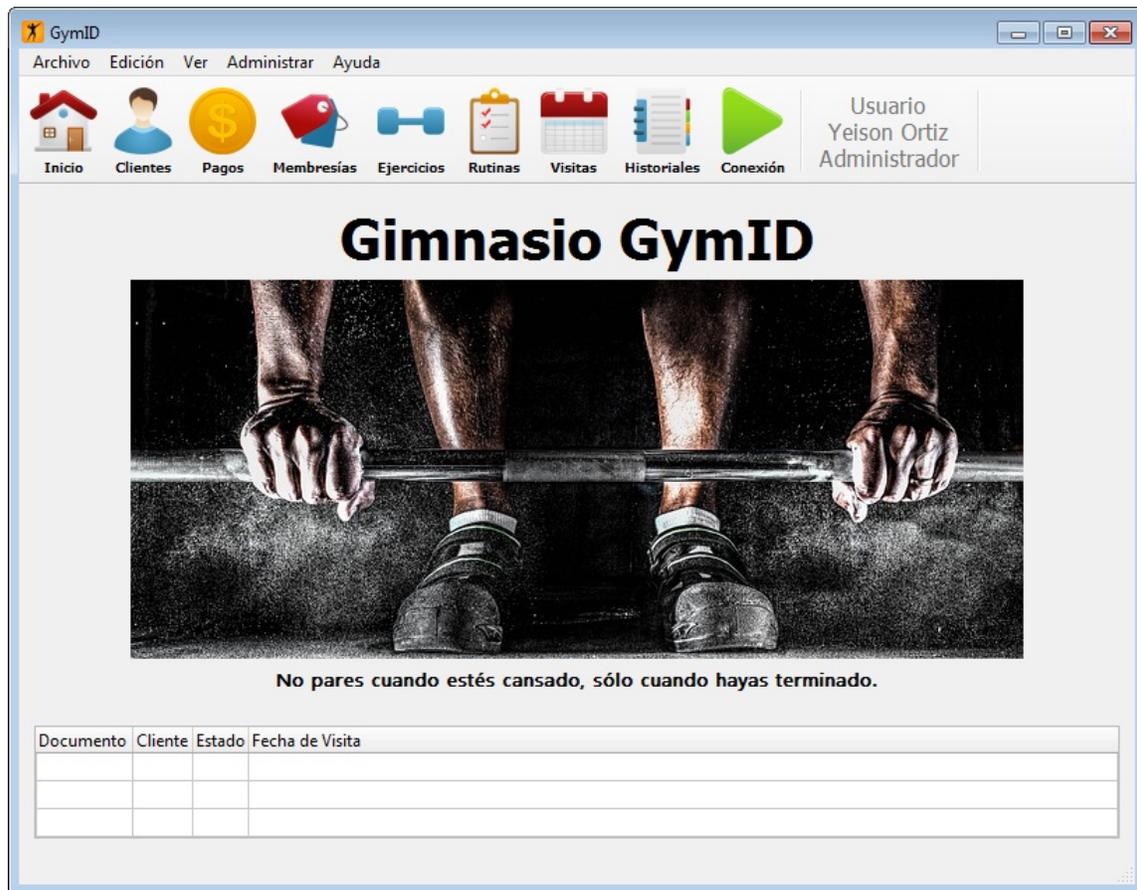


Figura 3.10: Interfaz de la Ventana Principal

pequeños (Inicio, Clientes, Pagos, Membresías, Ejercicios, Rutinas, Visitas, Historiales y Conexión) integra las funciones principales de la aplicación y se encuentra dividida por el tipo de función. Estas dos barras contienen funciones que serán habilitadas dependiendo del tipo de usuario que ingrese.

El Usuario Actual ubicado en la parte superior derecha muestra el usuario actual con el que se ha iniciado sesión, mostrando datos de interés como el nombre completo del usuario y su rol en la aplicación. Cualquier actividad realizada en la aplicación será guardada en la base de datos como un registro de actividad al Usuario Actual conectado.

La Información de Inicio ubicado en la parte central de la ventana es un espacio para ingresar información deliberada por el usuario de la aplicación. Principalmente fue diseñado para ingresar el nombre del gimnasio e información como dirección o teléfono, además de un logotipo o imagen que represente al gimnasio. Este espacio será mostrado solamente cuando se inicia sesión y a través de la función 'Inicio' ya que este espacio es utilizado para mostrar los menús de funciones y las tablas de la base de datos.

La Tabla de Visitas Recientes ubicado en la parte inferior muestra los ingresos más recientes de los clientes al gimnasio, es un atajo a las visitas realizadas, además muestra datos importantes como el estado de la cuenta y fecha y hora del ingreso.

3.5.3 Administrar Usuarios. Los usuarios son los empleados del gimnasio que se encargan de operar la aplicación GymID. Las funciones de usuario en la aplicación solo pueden ser utilizadas por un usuario Administrador y este podrá ver, crear y eliminar los usuarios de la base de datos.

Como se ha mencionado anteriormente existen 3 tipos de usuarios, el primero es el Administrador que es el encargado de administrar el gimnasio y por lo tanto tiene acceso a todas las funciones de la aplicación. La Figura 3.11 muestra los permisos del usuario Administrador.

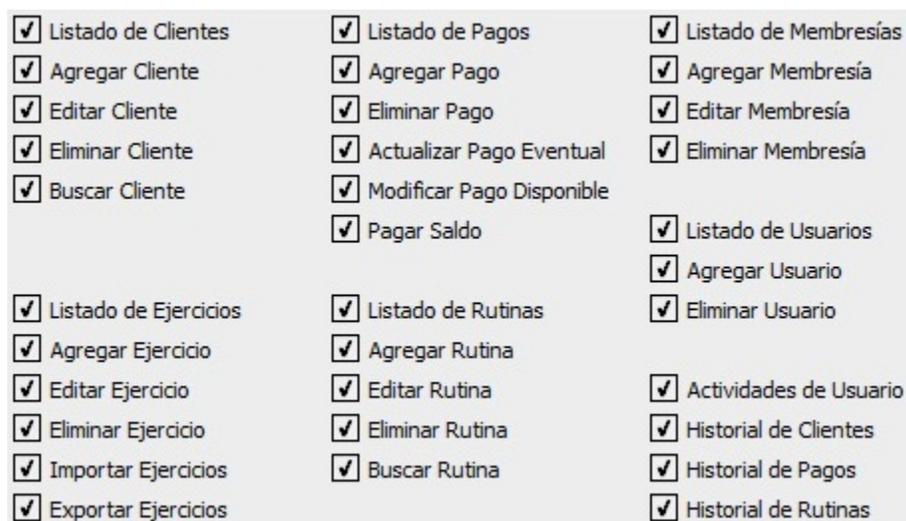


Figura 3.11: Permisos del Usuario Administrador

Otro usuario es el Recepcionista que se encarga de atender a los clientes en la recepción del gimnasio, este usuario tiene acceso a todas las funciones de clientes y pagos sin embargo no tiene accesos a la mayoría de funciones de ejercicios y rutinas. La Figura 3.12 muestra los permisos del usuario Recepcionista.

3.5. APLICACIÓN PARA ESCRITORIO

<input checked="" type="checkbox"/> Listado de Clientes	<input checked="" type="checkbox"/> Listado de Pagos	<input checked="" type="checkbox"/> Listado de Membresías
<input checked="" type="checkbox"/> Agregar Cliente	<input checked="" type="checkbox"/> Agregar Pago	<input type="checkbox"/> Agregar Membresía
<input checked="" type="checkbox"/> Editar Cliente	<input checked="" type="checkbox"/> Eliminar Pago	<input type="checkbox"/> Editar Membresía
<input checked="" type="checkbox"/> Eliminar Cliente	<input checked="" type="checkbox"/> Actualizar Pago Eventual	<input type="checkbox"/> Eliminar Membresía
<input checked="" type="checkbox"/> Buscar Cliente	<input checked="" type="checkbox"/> Modificar Pago Disponible	<input type="checkbox"/> Listado de Usuarios
	<input checked="" type="checkbox"/> Pagar Saldo	<input type="checkbox"/> Agregar Usuario
		<input type="checkbox"/> Eliminar Usuario
<input checked="" type="checkbox"/> Listado de Ejercicios	<input checked="" type="checkbox"/> Listado de Rutinas	<input checked="" type="checkbox"/> Actividades de Usuario
<input type="checkbox"/> Agregar Ejercicio	<input type="checkbox"/> Agregar Rutina	<input checked="" type="checkbox"/> Historial de Clientes
<input type="checkbox"/> Editar Ejercicio	<input type="checkbox"/> Editar Rutina	<input checked="" type="checkbox"/> Historial de Pagos
<input type="checkbox"/> Eliminar Ejercicio	<input type="checkbox"/> Eliminar Rutina	<input checked="" type="checkbox"/> Historial de Rutinas
<input type="checkbox"/> Importar Ejercicios	<input checked="" type="checkbox"/> Buscar Rutina	
<input type="checkbox"/> Exportar Ejercicios		

Figura 3.12: Permisos del Usuario Recepcionista

Por último el usuario Instructor se encarga de entrenar a los clientes del gimnasio, este casi no tiene permisos en la aplicación, pero tiene acceso a todas las funciones de ejercicios y rutinas de entrenamiento. La Figura 3.13 muestra los permisos del usuario Instructor.

<input checked="" type="checkbox"/> Listado de Clientes	<input type="checkbox"/> Listado de Pagos	<input checked="" type="checkbox"/> Listado de Membresías
<input type="checkbox"/> Agregar Cliente	<input type="checkbox"/> Agregar Pago	<input type="checkbox"/> Agregar Membresía
<input type="checkbox"/> Editar Cliente	<input type="checkbox"/> Eliminar Pago	<input type="checkbox"/> Editar Membresía
<input type="checkbox"/> Eliminar Cliente	<input type="checkbox"/> Actualizar Pago Eventual	<input type="checkbox"/> Eliminar Membresía
<input checked="" type="checkbox"/> Buscar Cliente	<input type="checkbox"/> Modificar Pago Disponible	<input type="checkbox"/> Listado de Usuarios
	<input type="checkbox"/> Pagar Saldo	<input type="checkbox"/> Agregar Usuario
		<input type="checkbox"/> Eliminar Usuario
<input checked="" type="checkbox"/> Listado de Ejercicios	<input checked="" type="checkbox"/> Listado de Rutinas	<input checked="" type="checkbox"/> Actividades de Usuario
<input checked="" type="checkbox"/> Agregar Ejercicio	<input checked="" type="checkbox"/> Agregar Rutina	<input checked="" type="checkbox"/> Historial de Clientes
<input checked="" type="checkbox"/> Editar Ejercicio	<input checked="" type="checkbox"/> Editar Rutina	<input type="checkbox"/> Historial de Pagos
<input checked="" type="checkbox"/> Eliminar Ejercicio	<input checked="" type="checkbox"/> Eliminar Rutina	<input checked="" type="checkbox"/> Historial de Rutinas
<input checked="" type="checkbox"/> Importar Ejercicios	<input checked="" type="checkbox"/> Buscar Rutina	
<input checked="" type="checkbox"/> Exportar Ejercicios		

Figura 3.13: Permisos del Usuario Instructor

3.5.4 Administrar Membresías. Las membresías son los tipos de inscripciones que existen en el gimnasio, se caracterizan por tener un precio y una duración. Las funciones de crear, editar y eliminar membresías solo pueden ser utilizadas por un

usuario Administrador, mientras que la función de ver listado de membresías puede ser usada por todos los usuarios. La Figura 3.14 muestra las funciones de las Membresías.

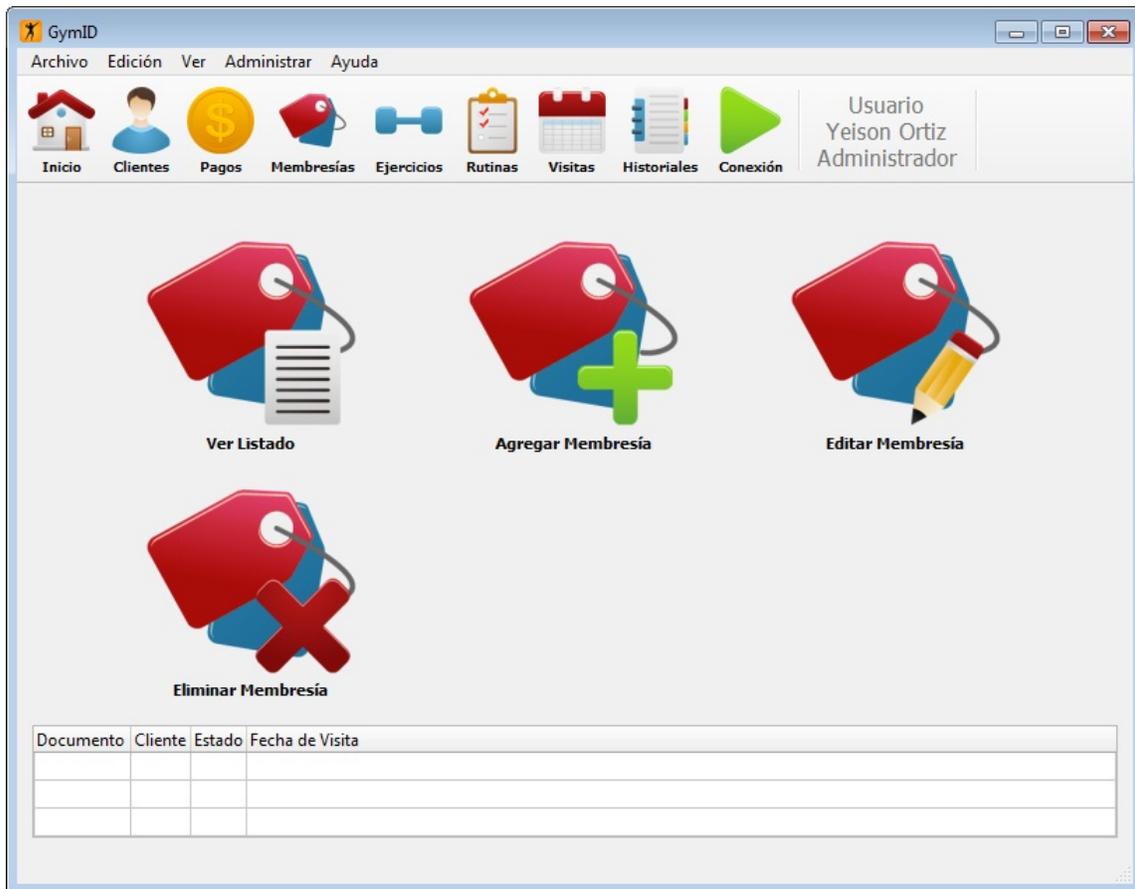


Figura 3.14: Menú de Membresías

Las membresías pueden ser cobradas de 2 formas. La primera es cuando la continuidad es Normal y el cobro se realiza por periodos de tiempo (1 mes, 1 semana, etc.), una vez termine ese periodo la membresía finaliza.

La segunda forma es cuando la continuidad es Adaptable y el cobro se realiza por los días asistidos en vez de periodos fijos de tiempo. Por ejemplo un cliente con continuidad Adaptable de 1 mes (equivalente a 30 días), gastará 1 día de su membresía cada vez que vaya al gimnasio, esto quiere decir que el periodo de activación de su membresía puede durar más de 1 mes. Sin embargo el plazo máximo para que el cliente gaste todos los días de su membresía tiene un límite de 4 veces el periodo de tiempo por el que pago.

La forma de pago Adaptable se implementa con el objetivo de innovar al darle mayor diversidad a las formas de pago de los gimnasios, llamando a un nuevo sector de clientela que por el trabajo o el estudio no pueden asistir continuamente al gimnasio. Al ser una

3.5. APLICACIÓN PARA ESCRITORIO

forma de pago flexible al cliente, lo más conveniente es que esta cueste un poco más que las formas de pago normales.

3.5.5 Administrar Clientes Los clientes son las personas que se inscriben al gimnasio, cada cliente puede guardarse con datos de la información personal y una foto que ayuda en la identificación del cliente. Las funciones de crear, editar y eliminar los clientes en la base de datos solo pueden ser utilizadas por los usuarios Administrador y Recepcionista, mientras que las funciones de ver listado y buscar clientes pueden ser usada por todos los usuarios. La Figura 3.15 muestra las funciones de los Clientes.

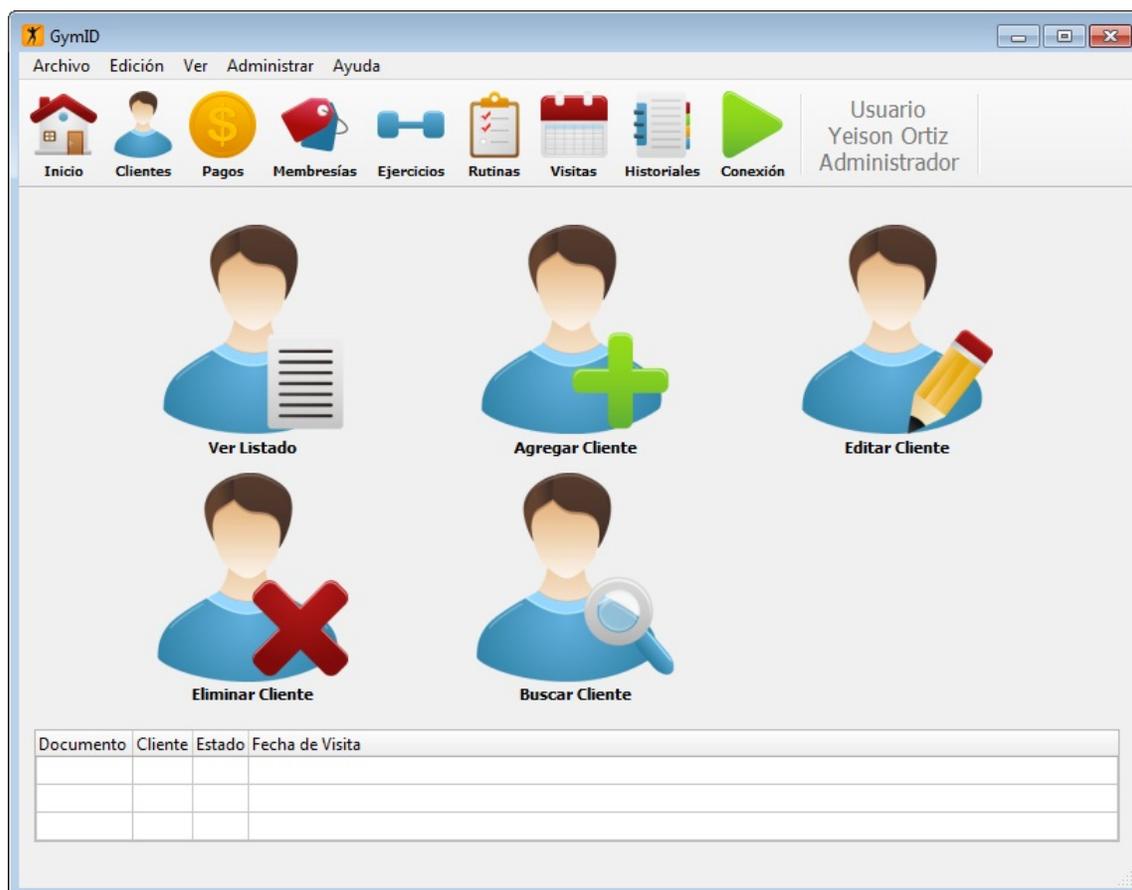


Figura 3.15: Menú de Clientes

Las fotografías de los clientes son indispensables para el buen funcionamiento del sistema porque hacen parte de la identificación, por esta razón GymID integra un sistema de captación de imágenes que permite al usuario tomar fotografías de los clientes por medio de una cámara web conectada, utilizando la función 'Tomar Fotografía' que hace parte de las funciones 'Agregar Cliente' y 'Editar Cliente'. La Figura 3.16 muestra

la interfaz gráfica para tomar fotografías a los clientes del gimnasio.

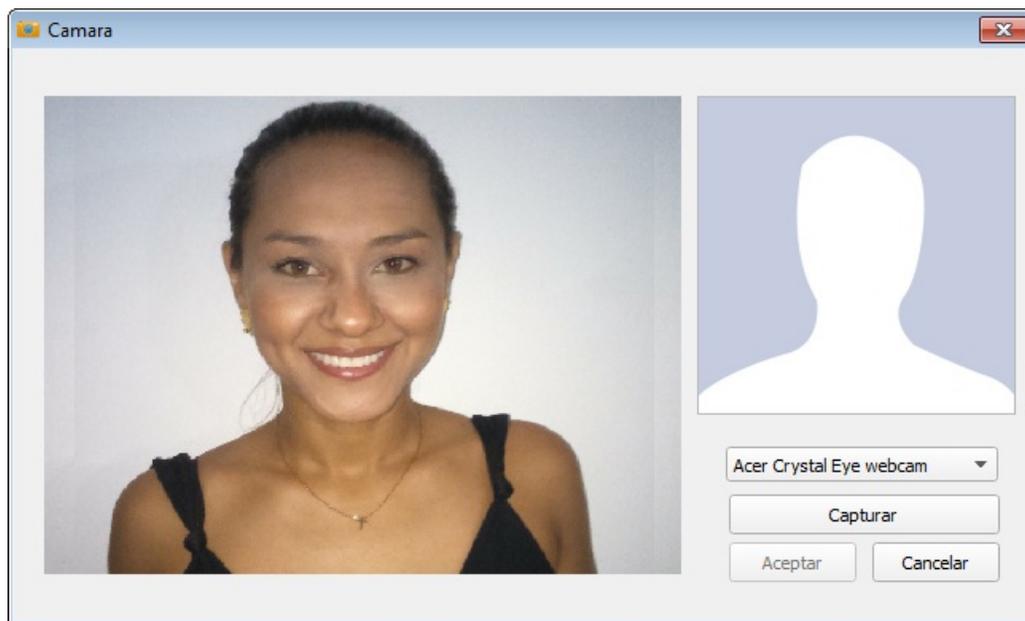


Figura 3.16: Interfaz para Tomar Fotografías

La aplicación puede reconocer cualquier cámara que se conecte al computador, incluso puede elegirse la cámara que desea utilizar, en caso de que hayan varias conectadas. Las imágenes capturadas no se guardan en el computador, estas son guardadas directamente en la base de datos para evitar almacenamiento innecesario del disco duro.

3.5.6 Administrar Pagos. Los pagos son cuentas de cobro de los clientes, cada pago se factura y se asigna al cliente que realiza el pago. Las funciones de pagos solo pueden ser utilizadas por los usuarios Administrador y Recepcionista, estos podrán ver, crear y eliminar los pagos de la base de datos, además los usuarios también podrán actualizar pagos eventuales, modificar pagos disponibles y pagar saldos. La Figura 3.17 muestra las funciones de los Pagos.

3.5. APLICACIÓN PARA ESCRITORIO

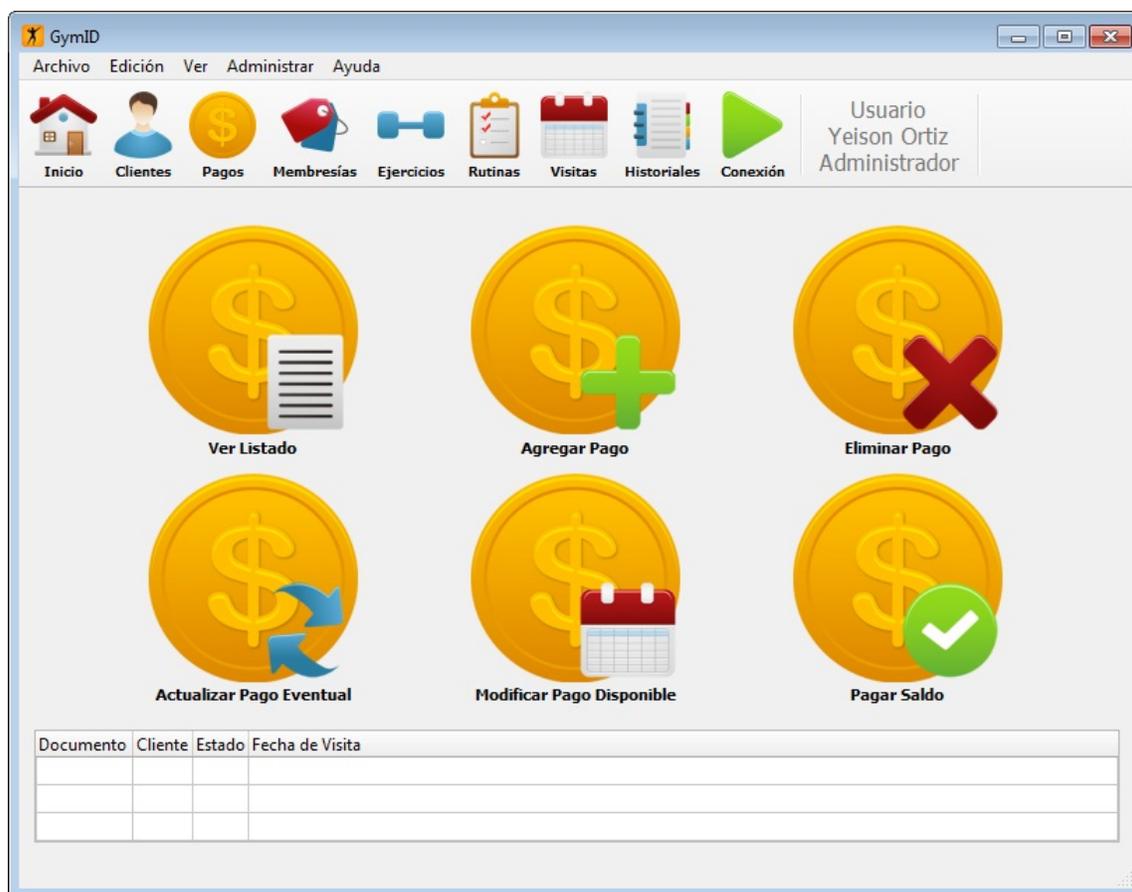


Figura 3.17: Menú de Pagos

Al realizarse un pago el cliente puede elegir las fechas de activación de ese pago, esto quiere decir que el cliente puede elegir el día que su cuenta iniciará su activación. La aplicación ofrece 3 tipos de activación:

- Inmediata: La fecha de inicio corresponde a la fecha del día en curso.
- Postergada: La fecha de inicio puede ser establecida a partir de la fecha del siguiente día del que se encuentra en curso.
- Eventual: No se establece fecha de inicio. El pago queda a la espera de que el usuario indique la fecha de inicio.

Los pagos son organizados por orden cronológico, esto quiere decir que el pago más reciente que ha realizado un cliente es el que tiene la fecha de inicio más lejana y por lo tanto será el último que consumirá si no realiza más pagos. Por lo tanto un usuario puede programar la fecha inicio de un nuevo pago únicamente después de la fecha final del último pago que haya realizado ese cliente.

3.5.7 Administrar Ejercicios. Los ejercicios son utilizados para crear las rutinas de entrenamiento del gimnasio, cada ejercicio puede ser organizado por zonas y músculos del cuerpo. Las funciones de crear, editar y eliminar ejercicios solo pueden ser utilizadas por los usuarios Administrador e Instructor, además los usuarios también podrán importar y exportar banco de datos de ejercicios por medio de las funciones en la pestaña ‘Edición’ de la barra de menú. Mientras que la función de ver listado puede ser usada por todos los usuarios. La Figura 3.18 muestra las funciones de los Ejercicios.

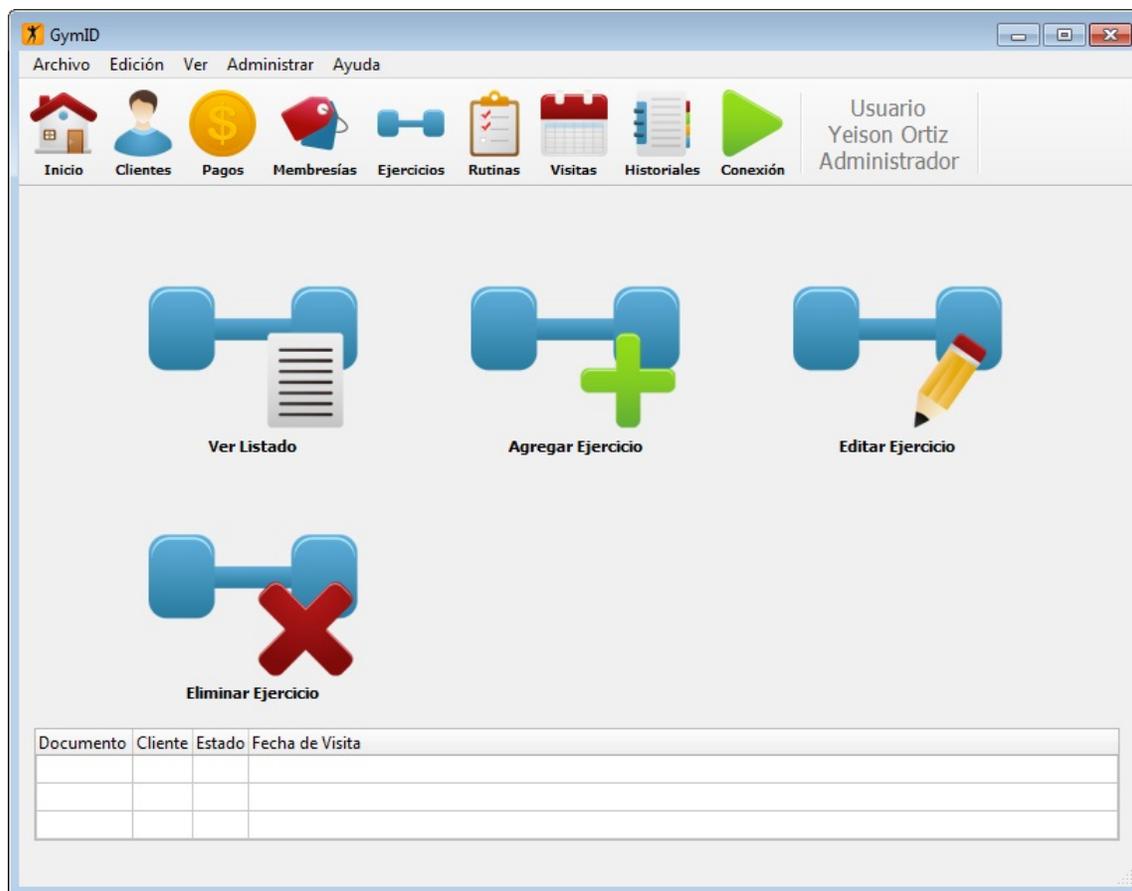


Figura 3.18: Menú de Ejercicios

La aplicación incluye 2 funciones que permitirán la importación y la exportación de ejercicios de la base de datos, estas funciones se realizan con el objetivo de facilitar la gestión de los ejercicios ya que el sistema no cuenta con ningún ejercicio almacenado por defecto en la base de datos, por lo tanto el usuario deberá ingresarlos por su cuenta. GymID no proporciona un banco de ejercicios por defecto porque da la libertad para que el usuario de la aplicación pueda personalizar su banco de datos, ya que un banco de datos predeterminado limitaría los servicios ofrecidos por un gimnasio, por el contrario, un gimnasio puede decidir que ejercicios instruir dependiendo de las maquinas que disponga o la utilización de ejercicios poco conocidos.

3.5. APLICACIÓN PARA ESCRITORIO

Sin embargo, GymID trae consigo un archivo (Ejercicios Everkinetic.txt) con 290 ejercicios con imágenes, basados en el banco de datos de la página web de Everkinetic (<http://db.everkinetic.com/>), que pueden ser importados exclusivamente a la base de datos de GymID por medio de la función 'Importar Ejercicios'. Además cuando el usuario quiera borrar la base de datos o reinstalar el sistema, podrá sacar una copia del banco de ejercicios almacenados por medio de la función 'Exportar Ejercicios'.

3.5.8 Administrar Rutinas. La rutina de entrenamiento es un plan de ejercicios físicos programados, cada rutina contiene información general del cliente y del entrenamiento. El entrenamiento se caracteriza por tener ejercicios aplicados al cliente con información de los días de la semana que se realizan, la intensidad, las series y las repeticiones. Las funciones de crear, editar y eliminar rutinas de entrenamiento solo pueden ser utilizadas por los usuarios Administrador e Instructor, mientras que la función de ver listado y buscar rutinas de entrenamiento pueden ser usada por todos los usuarios. La Figura 3.19 muestra las funciones de las Rutinas.

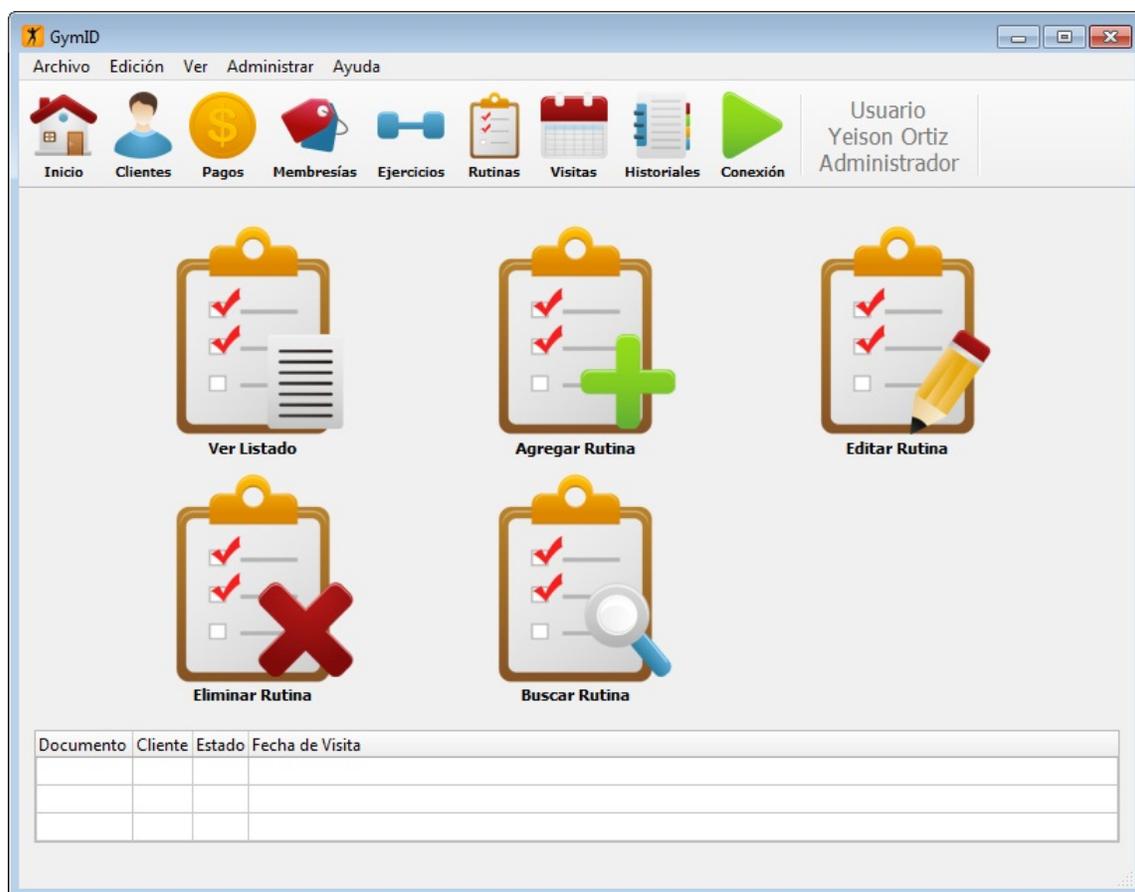


Figura 3.19: Menú de Rutinas

Las rutinas almacenadas en la base de datos pueden ser consultadas en cualquier momento por medio de la función ‘Buscar Rutina’, esta abrirá un perfil de la rutina con toda su información. La rutina abierta podrá ser generada con o sin imágenes de los ejercicios como un archivo PDF imprimible. La Figura 3.20 muestra la interfaz gráfica del perfil de la rutina de entrenamiento.

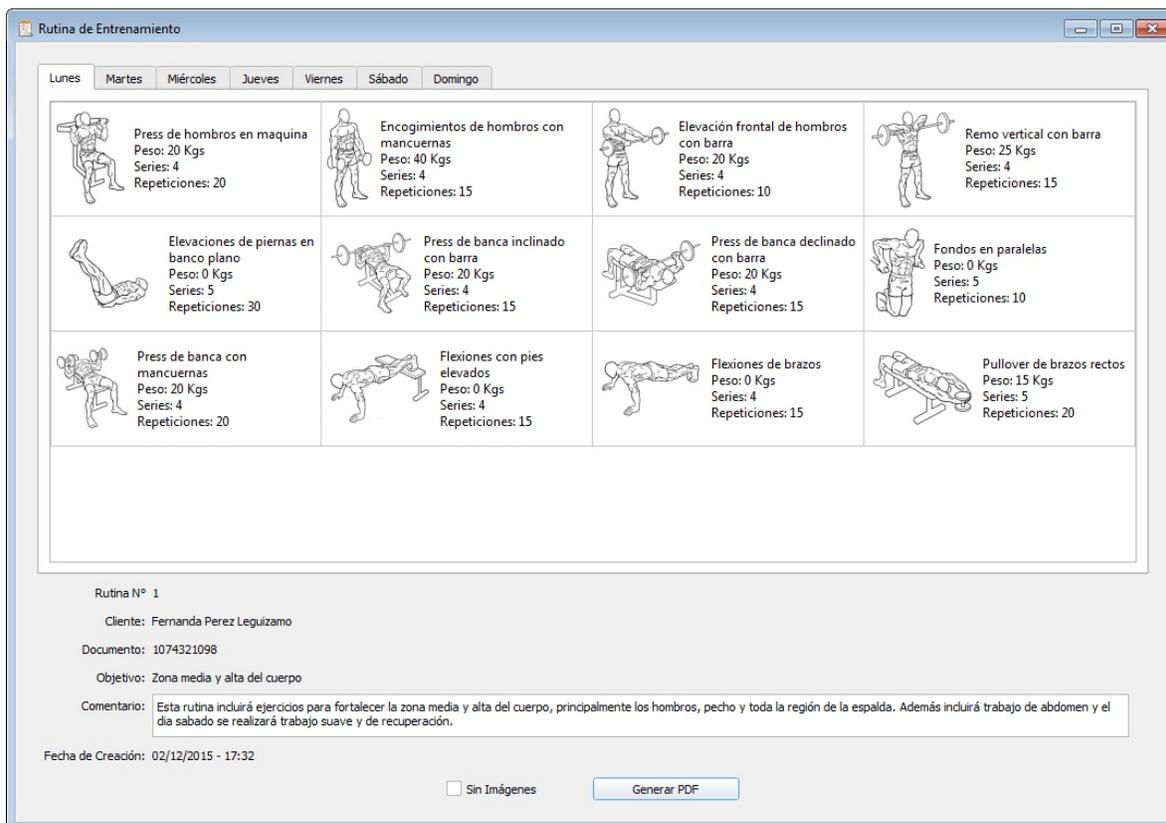


Figura 3.20: Interfaz del perfil Rutina

3.5.9 Historial de Registros. Los historiales muestran al usuario toda la información de las actividades de los usuarios, clientes, pagos y rutinas de entrenamiento de la base de datos. Todas las funciones pueden ser utilizadas por los usuarios Administrador y Recepcionista, mientras que el usuario Instructor solo tendrá acceso a los historiales de actividades de usuario, clientes y rutinas. La Figura 3.21 muestra las funciones de los Historiales.

Cuando se utiliza una función de historiales, la ventana principal mostrará una tabla con una lista de datos organizados por fechas y horas en que se realizaron. Si desea hacer un filtro a la tabla, la casilla desplegable de ‘Rango Temporal’ puede organizar la lista de acuerdo a un periodo de tiempo. El usuario también puede imprimir la información mostrada generando un archivo PDF por medio del botón ‘Generar PDF’.

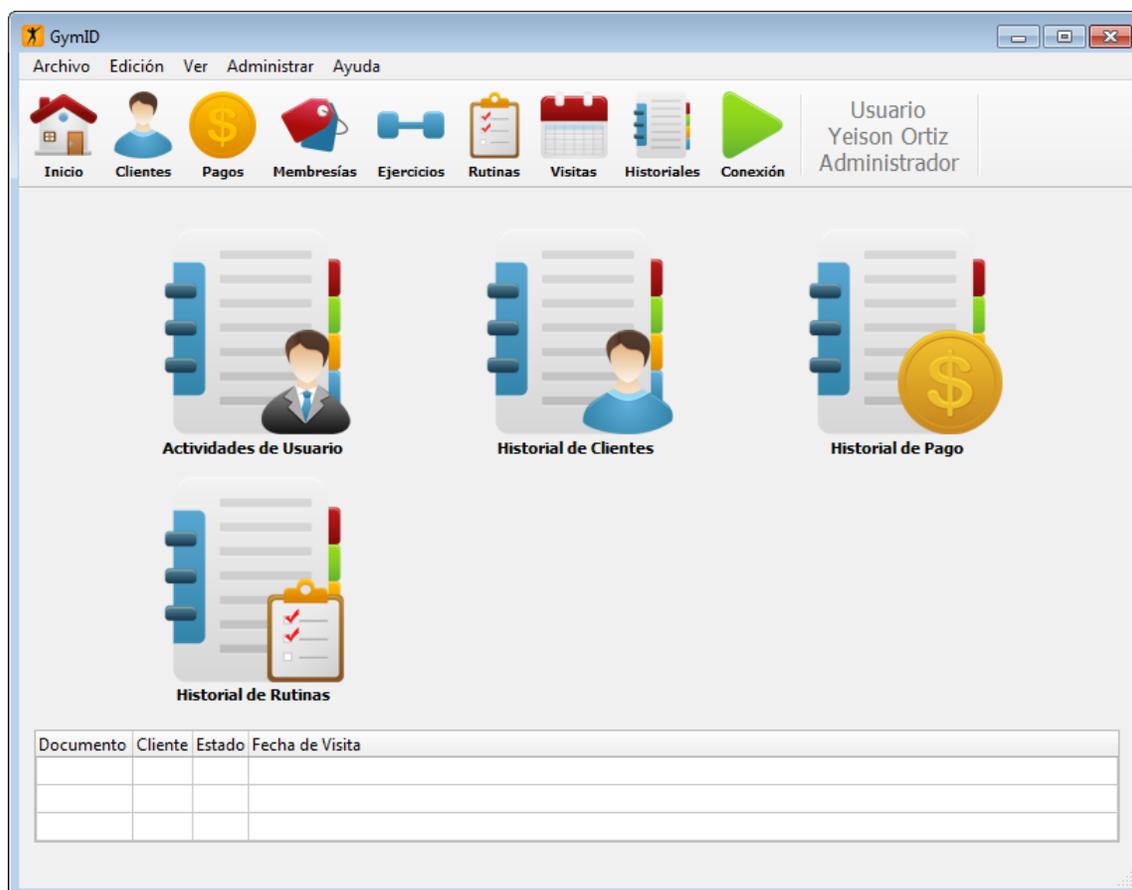


Figura 3.21: Menú de Historiales

3.5.10 Perfil del Cliente. Los perfiles pueden ser abiertos por medio de funciones de doble clic en tablas, pero la forma principal de abrirlos es desde el menú de clientes con la función 'Buscar Cliente'. La ventana de perfil muestra la información personal, datos de la cuenta y los pagos de un cliente determinado, también permite al usuario asignar etiquetas NFC al cliente. La Figura 3.22 muestra la interfaz gráfica del perfil del cliente.

La ventana de perfil muestra información completa sobre el cliente, en la parte izquierda se ubica la foto del perfil que le permite al usuario Recepcionista verificar la autenticidad del cliente al momento de identificarse en el sistema. En la parte superior derecha se ubica toda la información personal del cliente.

En la parte central derecha hay un botón que permite registrar manualmente una visita. Esta función es alternativa para el sistema lector de etiquetas NFC y requiere que la cuenta del cliente se encuentre activa. Por último en la parte inferior se ubica la información de la cuenta y del último pago que realizó el cliente.



Figura 3.22: Perfil del Cliente

Adicionalmente la ventana perfil incluye tres vistas divididas en pestañas que complementan la información y las funcionalidades del cliente. La pestaña de ‘Historial’ muestra información de los pagos que ha realizado el cliente y un registro de visitas al gimnasio. La Figura 3.23 muestra la interfaz gráfica del historial del cliente.

Mientras que la pestaña ‘Asignar Etiqueta’ permite al usuario asignar una etiqueta NFC de identificación al cliente. Esta función asigna una etiqueta NFC a un cliente del gimnasio en la base de datos. Las etiquetas asignadas a clientes pueden ser borradas en cualquier momento, solo puede asignarse una etiqueta por cliente y una etiqueta a la vez. La Figura 3.24 muestra la interfaz gráfica para asignar una etiqueta a un cliente.

3.5. APLICACIÓN PARA ESCRITORIO

Perfil de Fernanda Perez Leguizamo

HISTORIAL DE PAGOS

	Código	Descripción	Continuidad	Estado	Activación	Precio	Cantidad	Costo	Pago	Saldo	Fe
1	6	Semanal	Normal	Disponible	Postergada	8000	2	16000	16000	0	04,
2	5	Mensual Estudiante	Normal	Disponible	Postergada	25000	1	25000	25000	0	01,
3	2	Mensual Estudiante	Normal	Activo	Postergada	25000	1	25000	25000	0	25,

HISTORIAL DE VISITAS

	Documento	Cliente	Estado	Fecha y Hora
1	1074321098	Fernanda Perez Leguizamo	Activo	19/01/2016 - 04:45:29 pm
2	1074321098	Fernanda Perez Leguizamo	Activo	18/01/2016 - 07:12:14 pm
3	1074321098	Fernanda Perez Leguizamo	Activo	17/01/2016 - 07:11:03 pm
4	1074321098	Fernanda Perez Leguizamo	Activo	16/01/2016 - 07:01:30 pm
5	1074321098	Fernanda Perez Leguizamo	Activo	15/01/2016 - 07:01:06 pm
6	1074321098	Fernanda Perez Leguizamo	Activo	14/01/2016 - 06:59:00 pm
7	1074321098	Fernanda Perez Leauizamo	Activo	13/01/2016 - 06:58:38 pm

Perfil Historial Asignar Etiqueta

Figura 3.23: Historial del Cliente

Perfil de Fernanda Perez Leguizamo

Bienvenido al asistente de asignación de etiqueta, por favor escoja el tipo de etiqueta correspondiente y proceda a iniciar lectura de la etiqueta.

Lectura de Etiquetas

Tipo de Etiqueta:

Para obtener el código único de identificación que provee la etiqueta haga clic en iniciar lectura y acerque brevemente la etiqueta al lector.



Seleccione el tipo de etiqueta, y luego haga clic en 'Iniciar Lectura'.

Perfil Historial Asignar Etiqueta

Figura 3.24: Asignar Etiqueta al Cliente

3.5.11 Identificar. Las visitas son las asistencias que realiza un cliente al gimnasio, cada visita es almacenada en la base de datos como un registro de actividad del cliente. La visita que se registra bajo una membresía adaptable se guarda en la base de datos y disminuye la cantidad de visitas restantes del cliente en 1 día, el cliente puede registrar varias visitas un mismo día pero únicamente se disminuirá en 1 la cantidad de visitas restantes por cada día. En cambio, la visita que registra un cliente con membresía normal no altera la cantidad de visitas restantes ya que estas dependen únicamente del transcurso de los días. Las visitas registradas siempre muestran en la aplicación el número de documento y el nombre completo del cliente, el estado de la cuenta y la fecha con la hora que se realiza la visita.

Cuando un cliente realiza una visita, primero pasa la etiqueta NFC por el lector, el lector adquiere la ID de la etiqueta y la envía a la aplicación GymID. La aplicación busca la información de la cuenta del cliente y la envía al lector de etiquetas NFC para que sea mostrada en la pantalla LCD. La Figura 3.25 muestra la identificación del cliente activo en el dispositivo lector de etiquetas NFC.



Figura 3.25: Identificación de Cliente Activo

Una vez se ha mostrado el mensaje de bienvenida en la pantalla LCD, si la cuenta esta activa se muestra organizadamente el nombre completo del cliente, los días restantes del pago activo, la fecha que inicio la activación del pago y la fecha que finaliza la

3.5. APLICACIÓN PARA ESCRITORIO

activación. Simultáneamente la aplicación agrega una visita al cliente en la base de datos y la tabla de visitas recientes de la aplicación mostrará la visita en color verde. La Figura 3.26 muestra la identificación del cliente activo en la aplicación GymID.

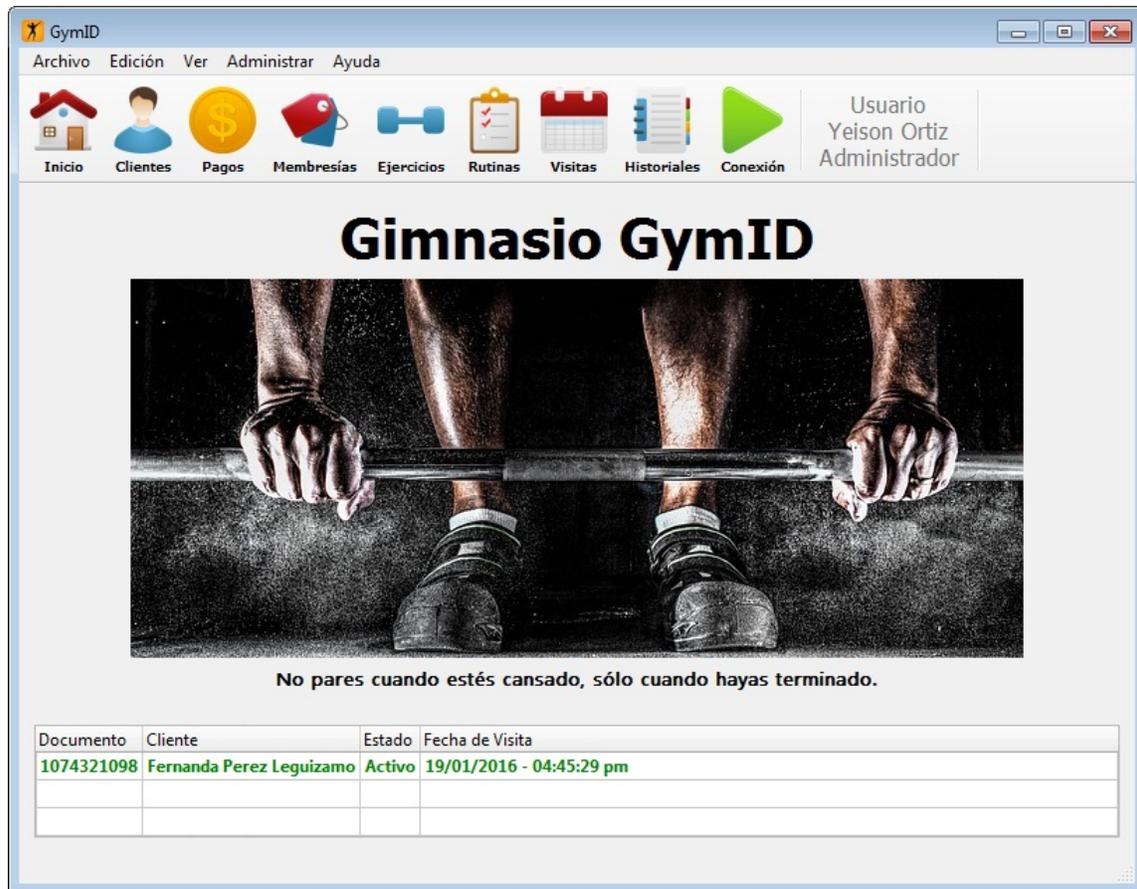


Figura 3.26: Identificación de Cliente Activo en GymID

En caso de que la cuenta de un cliente se encuentre Inactiva, el dispositivo lector mostrará al cliente el estado actual de su cuenta con la fecha de finalización del último pago activado. La Figura 3.27 muestra la identificación del cliente inactivo en el dispositivo lector de etiquetas NFC. Mientras que GymID advertirá al Recepcionista de la inactividad del cliente que realice la visita, estableciendo la visita en color rojo en la tabla de visitas recientes. La Figura 3.28 muestra la identificación del cliente inactivo en la aplicación GymID.

En cualquier momento el usuario de la aplicación puede revisar el listado de visitas por medio de la función 'Visitas', ubicado en la barra de herramientas de la ventana principal. La información es mostrada en la interfaz gráfica de los listados, con la posibilidad de realizar filtros al listado y abrir perfiles de clientes con la función de doble clic.



Figura 3.27: Identificación de Cliente Inactivo

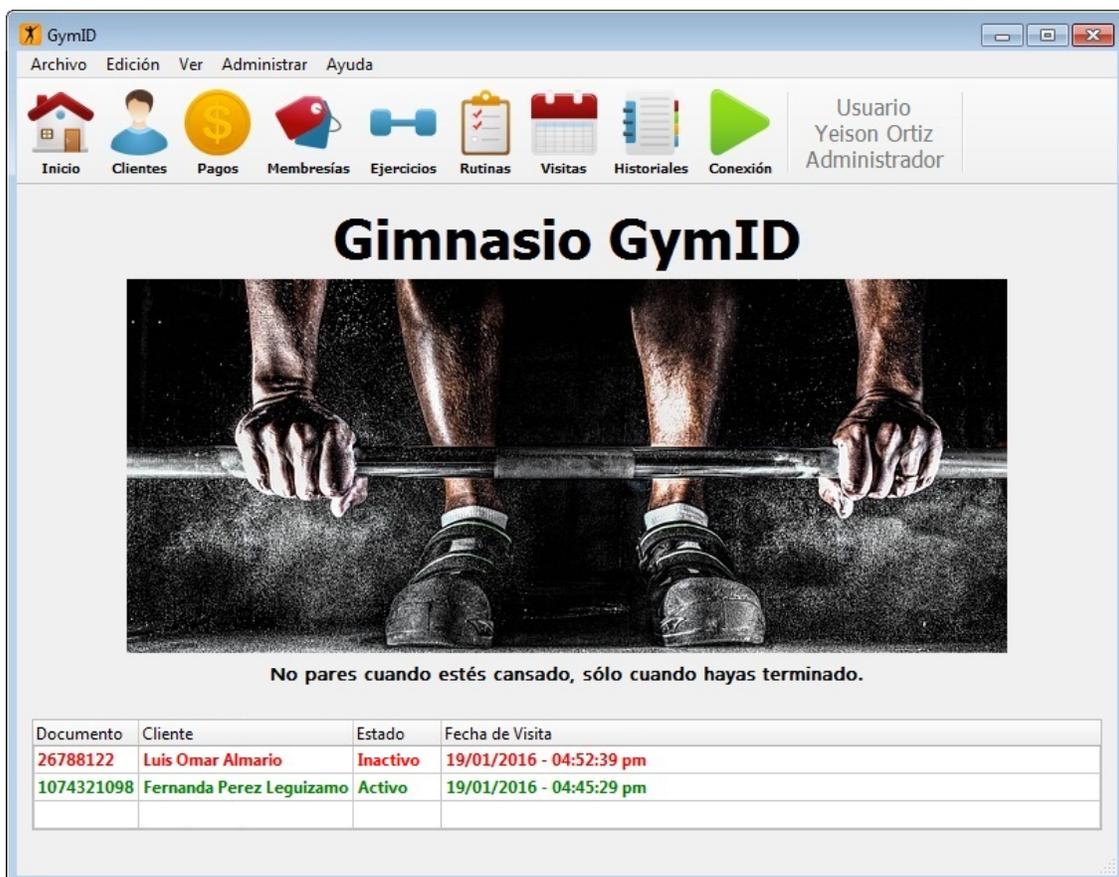


Figura 3.28: Identificación de Cliente Inactivo en GymID

4. CONCLUSIONES

La aplicación de la tecnología NFC en los sistemas de identificación, mejora la velocidad, la seguridad y facilita los procesos de autenticación en comparación con otras tecnologías que pueden llegar a ser más susceptible a errores, como los sistemas biométricos.

La tecnología NFC es una tecnología inalámbrica de corto alcance, lo cual supera la principal falencia de seguridad que tiene la tecnología RFID en los sistemas de identificación.

La aplicación para escritorio GymID provee una gran variedad de funciones al usuario soportado por una completa base de datos que optimizan el sistema permitiendo a cualquier empresa mediana o pequeña competir con las grandes empresas de gimnasios.

El sistema permite la implementación de un nuevo método de pago en el gimnasio, que le ofrece al cliente la posibilidad de pagar por los días asistidos, aumentando los servicios ofrecidos por el gimnasio y atrayendo nuevos clientes que por diversos motivos no puedan asistir continuamente al gimnasio.

La implementación de la aplicación para escritorio como software libre, permite el mejoramiento continuo para el futuro del proyecto desarrollado con adaptaciones a sistemas diversos.

El lenguaje de modelado facilita el diseño e implementación del sistema a nivel de software y permite ver claramente la relación con los componentes de hardware involucrados en el sistema.

5. RECOMENDACIONES

El registro manual de clientes solo debe ser utilizado en casos especiales, ya sea que el cliente olvidó o perdió su etiqueta de identificación o el lector de etiquetas NFC no funcione correctamente.

El sistema puede implementarse como un control de acceso automatizado, sin embargo, es necesario que exista una supervisión rigurosa de las operaciones realizadas, ya que una etiqueta NFC podría ser utilizada por varias personas y el sistema debería ser capaz de identificar el verdadero dueño de la etiqueta.

El lector puede ser implementado para interactuar con otros módulos NFC, esto podría permitir la utilización de teléfonos móviles en vez de tarjetas NFC, sin embargo, se debe crear una aplicación para celulares que permita ya sea la interfaz de comunicación y protocolo (NFCIP-1) entre los dispositivos activos o la emulación de etiquetas NFC por parte del modulo del teléfono en el modo de operación como tarjeta.

También puede aprovecharse la memoria EEPROM de las tarjetas de identificación NFC para guardar información relacionada con el cliente, por ejemplo agregar información de pagos realizados o rutinas de entrenamiento del cliente para luego puedan ser visualizadas desde un dispositivo móvil o una interfaz del gimnasio. Además, a este sistema se pueden agregar dispositivos biomédicos que supervisen la actividad cardíaca y física del cliente.

El proyecto puede ser modificado para ser adaptado a otros sistemas de identificación con gestión de base de datos, por ejemplo en la entrada de un parque de diversiones, cine o transporte publico donde se gestione la entrada por medio de tarjetas de identificación soportado por una base de datos que almacene las cuentas de los clientes y los pagos. Igualmente la identificación de pacientes en hospitales donde se lleve un registro de la información de los pacientes con su respectiva historia clínica.

Este proyecto está protegido bajo la licencia de código abierto LGPL y cualquier trabajo derivado de este proyecto, ya sea copia, modificación o distribución debe incluirse bajo la misma licencia de software libre, reconociendo a los autores originales.

BIBLIOGRAFÍA

ANÓNIMO. Aprenda Qt4 desde hoy mismo. Documento Libre, 2010. 120 p.

CERNÁNDES GÓMEZ, Harry Alejandro; ZAPATA RAMÍREZ, Elmer Kristopher. Identificación de personas mediante el reconocimiento dactilar y su aplicación a la seguridad organizacional. Trabajo de grado de ingeniero de sistemas. Lima: Universidad Nacional Mayor de San Marcos. Facultad de ingeniería de sistemas e informática, Área de ingeniería de sistemas, 2006. 171 p.

Limor ‘Ladyada’ Fried. Adafruit PN532 RFID/NFC Breakout and Shield [en línea]. En: Adafruit. 12, 2012. no. 1, p. 53. Disponible en Internet: <https://learn.adafruit.com/downloads/pdf/adafruit-pn532-rfid-nfc.pdf>

NAVÍO RECIO, Juan Luis. Movilidad: Identificación por Radiofrecuencia (RFID). Trabajo de grado de ingeniero de telecomunicaciones. Sevilla: Universidad de Sevilla. Departamento de Sistemas y Automática, Área de ingeniería Telemática. 2006. 194 p.

NFC en el móvil: cómo, cuándo y para qué [en línea]. En: Xataka Android. Febrero, 2014. Disponible en Internet: <http://www.xatakandroid.com/mundogalaxy/nfc-en-el-movil-como-cuando-y-para-que>

PASTOR, Javier. Siete cosas que puedes hacer gracias a la tecnología NFC [en línea]. En: Xataka Móvil. Marzo, 2013. Disponible en Internet: <http://www.xatakamovil.com/futuro/7-cosas-que-puedes-hacer-gracias-a-la-tecnologia-nfc>

PENALVA, Javier. NFC: qué es y para qué sirve [en línea]. En: Xataka. Enero, 2011. Disponible en Internet: <http://www.xataka.com/moviles/nfc-que-es-y-para-que-sirve>

PN532/C1 [en línea]: Product data sheet. NXP Semiconductors, 2007. Disponible en Internet: <https://www.adafruit.com/datasheets/pn532longds.pdf>

Qt for Windows - Deployment [en línea]. Qt company. Disponible en Internet: <http://doc.qt.io/qt-5/windows-deployment.html>

Qué son los sistemas de gestión empresarial? [en línea]. Buenos Aires: Informática hoy. Disponible en Internet: <http://www.informatica-hoy.com.ar/aprender-informatica/Que-son-los-sistemas-de-gestion-empresarial.php>

RUMBAUGN, James; JACOBSON Ivar, BOOCH Grady. El lenguaje unificado de modelado: Manual de referencia. 2 ed. Madrid: Pearson Education, 2007. 688 p.

SCHMULLER, Joseph. Aprendiendo UML en 24 horas. Juárez: Pearson Educación Latinoamerica. 404 p.

THRASHER, James. RFID vs. NFC: What's the Difference? [en línea]. En: RFID insider. Octubre, 2013. Disponible en Internet: <http://blog.atlasrfidstore.com/rfid-vs-nfc>

W. EVANS, Brian. Arduino Notebook: A Beginner's Reference. San Francisco: Creative Commons, 2007. 70 p.

ANEXOS

A. Especificaciones de los Casos de Uso

En este anexo se presentan las tablas correspondientes a las especificaciones de cada caso de uso organizado por actores.

El Cuadro A.1 muestra el significado de las abreviaturas utilizadas en las especificaciones de los casos de uso.

Elemento	Significado
A	Act.1 Administrador del Gimnasio
C	Act.2 Cliente del Gimnasio
I	Act.3 Instructor del Gimnasio
R	Act.4 Recepcionista del Gimnasio
U	Usuario del Software
GUI	Interfaz Gráfica de Usuario
BD	Base de Datos
DNI	Numero del Documento Nacional de Identidad
ID	Numero de Identificación NFC
LCD	Pantalla LCD

Cuadro A.1: Diccionario de elementos del sistema

A.1. Casos de uso para el Administrador

Caso de Uso	CU.1 Administrar Membresía
Actor	Act.1 Administrador del Gimnasio (A)
Descripción	El caso de uso permite al Administrador crear, editar y eliminar las Membresías.
Flujo Básico	El caso de uso INICIA cuando el Administrador oprime el botón ‘Membresías’ desde su interfaz gráfica. 1. GUI muestra las opciones que puede realizar el A en la pantalla. 2. A Elige una opción en la interfaz gráfica. FIN del caso de uso.
Flujos Alternos	
Pre-condiciones	1. El U debe estar logueado con una cuenta de Administrador.
Post-condiciones	
Punto de extensión	2.1. A Elige Crear Membresía, hacer «extend»CU.2 Crear Membresía 2.2. A Elige Editar Membresía, hacer «extend»CU.3 Editar Membresía 2.3. A Elige Eliminar Membresía, hacer «extend»CU.4 Eliminar Membresía

Cuadro A.2: Especificación del caso de uso Administrar Membresía

A.1. CASOS DE USO PARA EL ADMINISTRADOR

Caso de Uso	CU.2 Crear Membresía
Actor	Act.1 Administrador del Gimnasio (A)
Descripción	El caso de uso permite al Administrador crear Membresías de cobro.
Flujo Básico	<p>El caso de uso INICIA cuando el Administrador del Gimnasio oprime el botón ‘Crear Membresía’ desde la ventana ‘Membresías’.</p> <ol style="list-style-type: none"> 1. GUI abre la ventana de asistente de creación de membresías. 2. A llena los datos de la nueva membresía. 3. A oprime el botón ‘Crear’. 4. GUI almacena los datos de la nueva membresía en la BD y cierra la ventana. <p>FIN del caso de uso.</p>
Flujos Alternos	<p>1. Campos relevantes no fueron completados</p> <p>En el paso 3 el Administrador oprime ‘Crear’ sin completar algunos campos.</p> <p>GUI informa al A que algunos campos relevantes no fueron completados.</p> <p>GUI muestra que campos debe llenar.</p> <p>Se repiten las acciones descritas a partir del paso 2.</p>
Pre-condiciones	<ol style="list-style-type: none"> 1. El U debe estar logueado con una cuenta de Administrador.
Post-condiciones	<ol style="list-style-type: none"> 1. GUI informa al A que la membresía ha sido creada. 2. GUI retorna a la ventana principal.
Punto de extensión	

Cuadro A.3: Especificación del caso de uso Crear Membresía

Caso de Uso	CU.3 Editar Membresía
Actor	Act.1 Administrador del Gimnasio (A)
Descripción	El caso de uso permite al Administrador editar Membresías de cobro.
Flujo Básico	<p>El caso de uso INICIA cuando el Administrador del Gimnasio oprime el botón ‘Editar Membresía’ desde la ventana ‘Membresías’.</p> <ol style="list-style-type: none"> 1. GUI abre la ventana de asistente de edición de membresías. 2. A escribe el código de la membresía que desea editar. 3. GUI carga los datos actuales de la membresía especificada. 4. A Edita los valores y oprime el botón ‘Guardar’. 5. GUI almacena los datos de la membresía en la BD y cierra la ventana. <p>FIN del caso de uso.</p>
Flujos Alternos	<ol style="list-style-type: none"> 1. Membresía no ha sido encontrada <p>En el paso 3 la GUI no encuentra la membresía con el código especificado por el Administrador. GUI informa al A que la membresía no existe. Se repiten las acciones descritas a partir del paso 2.</p>
Pre-condiciones	<ol style="list-style-type: none"> 1. El U debe estar logueado con una cuenta de Administrador.
Post-condiciones	<ol style="list-style-type: none"> 1. GUI informa al A que la membresía ha sido editada. 2. GUI retorna a la ventana principal.
Punto de extensión	

Cuadro A.4: Especificación del caso de uso Editar Membresía

A.1. CASOS DE USO PARA EL ADMINISTRADOR

Caso de Uso	CU.4 Eliminar Membresía
Actor	Act.1 Administrador del Gimnasio (A)
Descripción	El caso de uso permite al Administrador eliminar Membresías de cobro.
Flujo Básico	<p>El caso de uso INICIA cuando el Administrador del Gimnasio oprime el botón ‘Eliminar Membresía’ desde la ventana ‘Membresías’.</p> <ol style="list-style-type: none"> 1. GUI abre la ventana de asistente de eliminación de membresías. 2. A selecciona la membresía que desea eliminar. 3. A oprime el botón ‘Eliminar’. 4. GUI elimina los datos de la membresía seleccionada en la BD y cierra la ventana. <p>FIN del caso de uso.</p>
Flujos Alternos	<p>1. Ninguna membresía ha sido seleccionada</p> <p>En el paso 3 el Administrador oprime ‘Eliminar’ sin seleccionar la membresía.</p> <p>GUI informa al A que ninguna membresía fue seleccionada.</p> <p>Se repiten las acciones descritas a partir del paso 2.</p>
Pre-condiciones	1. El U debe estar logueado con una cuenta de Administrador.
Post-condiciones	<ol style="list-style-type: none"> 1. GUI informa al A que la membresía ha sido eliminada. 2. GUI retorna a la ventana principal.
Punto de extensión	

Cuadro A.5: Especificación del caso de uso Eliminar Membresía

Caso de Uso	CU.5 Administrar Usuario
Actor	Act.1 Administrador del Gimnasio (A)
Descripción	El caso de uso permite al Administrador crear y eliminar los Usuarios del programa.
Flujo Básico	El caso de uso INICIA cuando el Administrador oprime el botón 'Usuarios' desde su interfaz gráfica. 1. GUI muestra las opciones que puede realizar el A en la pantalla. 2. A Elige una opción en la interfaz gráfica. FIN del caso de uso.
Flujos Alternos	
Pre-condiciones	1. El U debe estar logueado con una cuenta de Administrador.
Post-condiciones	
Punto de extensión	2.1. A Elige Crear Usuario, hacer «extend»CU.6 Crear Usuario 2.2. A Elige Eliminar Usuario, hacer «extend»CU.7 Eliminar Usuario

Cuadro A.6: Especificación del caso de uso Administrar Usuario

A.1. CASOS DE USO PARA EL ADMINISTRADOR

Caso de Uso	CU.6 Crear Usuario
Actor	Act.1 Administrador del Gimnasio (A)
Descripción	El caso de uso permite al Administrador crear cuentas de Usuarios.
Flujo Básico	<p>El caso de uso INICIA cuando el Administrador del Gimnasio oprime el botón ‘Crear Usuario’ desde la ventana ‘Usuarios’.</p> <ol style="list-style-type: none"> 1. GUI abre la ventana de asistente de creación de U. 2. A llena los datos de el nuevo U. 3. A oprime el botón ‘Crear’. 4. GUI almacena los datos de el nuevo U en la BD y cierra la ventana. <p>FIN del caso de uso.</p>
Flujos Alternos	<p>1. Campos relevantes no fueron completados</p> <p>En el paso 3 el Administrador oprime ‘Crear’ sin completar algunos campos.</p> <p>GUI informa al A que algunos campos relevantes no fueron completados.</p> <p>GUI muestra que campos debe llenar.</p> <p>Se repiten las acciones descritas a partir del paso 2.</p>
Pre-condiciones	<ol style="list-style-type: none"> 1. El U debe estar logueado con una cuenta de Administrador.
Post-condiciones	<ol style="list-style-type: none"> 1. GUI informa al A que el U ha sido creado. 2. GUI retorna a la ventana principal.
Punto de extensión	

Cuadro A.7: Especificación del caso de uso Crear Usuario

A.1. CASOS DE USO PARA EL ADMINISTRADOR

Caso de Uso	CU.7 Eliminar Usuario
Actor	Act.1 Administrador del Gimnasio (A)
Descripción	El caso de uso permite al Administrador eliminar cuentas de Usuarios.
Flujo Básico	<p>El caso de uso INICIA cuando el Administrador del Gimnasio oprime el botón ‘Eliminar Usuario’ desde la ventana ‘Usuarios’.</p> <ol style="list-style-type: none"> 1. GUI abre la ventana de asistente de eliminación de U. 2. A escribe el DNI de el U que desea eliminar. 3. GUI muestra información de el U descrito por el DNI. 4. A oprime el botón ‘Eliminar’. 5. GUI elimina los datos de el U seleccionado en la BD y cierra la ventana. <p>FIN del caso de uso.</p>
Flujos Alternos	<p>1. Usuario no ha sido encontrado</p> <p>En el paso 3 la GUI no encuentra el Usuario con el DNI especificado por el Administrador.</p> <p>GUI informa al A que el U no existe.</p> <p>Se repiten las acciones descritas a partir del paso 2.</p>
Pre-condiciones	<ol style="list-style-type: none"> 1. El U debe estar logueado con una cuenta de Administrador.
Post-condiciones	<ol style="list-style-type: none"> 1. GUI informa al A que el U ha sido eliminado. 2. GUI retorna a la ventana principal.
Punto de extensión	

Cuadro A.8: Especificación del caso de uso Eliminar Usuario

A.1. CASOS DE USO PARA EL ADMINISTRADOR

Caso de Uso	CU.8 Generar PDF de historial de registros
Actor	Act.1 Administrador del Gimnasio (A)
Descripción	El caso de uso permite al Administrador organizar historiales (Clientes, Pagos, Actividades, etc) de la Base de Datos en un archivo de formato PDF.
Flujo Básico	<p>El caso de uso INICIA cuando el Administrador oprime el Botón ‘Historiales’ desde su interfaz gráfica.</p> <ol style="list-style-type: none"> 1. GUI muestra en la pantalla los historiales disponibles que A puede generar. 2. A selecciona los historiales que quiere en el reporte. 3. A presiona el botón ‘Generar PDF’. 4. GUI crea un archivo de formato PDF con la lista de historiales. <p>FIN del caso de uso.</p>
Flujos Alternos	<p>1. Ningún historial se ha seleccionado</p> <p>En el paso 3 el Administrador presiona el botón sin haber seleccionado ningún historial.</p> <p>GUI informa a A que no ha seleccionado ningún historial.</p> <p>Se repiten las acciones descritas a partir del paso 1.</p>
Pre-condiciones	1. El U debe estar logueado con una cuenta de Administrador.
Post-condiciones	<ol style="list-style-type: none"> 1. GUI informa al A que el PDF ha sido generado. 2. GUI retorna a la ventana principal.
Punto de extensión	

Cuadro A.9: Especificación del caso de uso Generar PDF de historial de registros

A.2. Casos de uso para el Cliente

Caso de Uso	CU.9 Identificar
Actor	Act.2 Cliente (C), Act.4 Recepcionista (R)
Descripción	El caso de uso permite la identificación del Cliente.
Flujo Básico	<p>El caso de uso INICIA cuando el Cliente pasa la etiqueta por el lector NFC.</p> <ol style="list-style-type: none"> 1. El Sistema adquiere el ID de la etiqueta. 2. GUI busca en la BD el C asignado a esa ID. 3. GUI muestra el perfil del C encontrado. 4. R verifica que el estado del C este activo. 5. R compara la foto del perfil con el C. 6. R permite el acceso al C. <p>FIN del caso de uso.</p>
Flujos Alternos	<p>1. ID no ha sido asignada a ningún Cliente En el paso 2 no se encuentra ningún Cliente con la ID leída. GUI informa al R que la ID no ha sido asignada. El LCD muestra al C que no ha sido encontrado. El caso de uso finaliza.</p> <p>2. El estado del Cliente no esta activo En el paso 4 el estado del Cliente no esta activo. R advierte al C que su estado no esta activo y no permite su acceso. El caso de uso finaliza.</p> <p>3. La foto del perfil no coincide con el Cliente En el paso 5 la foto del perfil no coincide con la imagen del Cliente. R pide el DNI del C. R compara datos importantes como numero del DNI, nombre completo. Si los datos coinciden, hacer CU.28 Tomar Fotografías y R permite el acceso al C.</p>
Pre-condiciones	1. Las etiquetas deben ser NFC.
Post-condiciones	1. El Sistema cambia de estado "a la espera de una nueva etiqueta".
Punto de extensión	

Cuadro A.10: Especificación del caso de uso Identificar

A.2. CASOS DE USO PARA EL CLIENTE

Caso de Uso	CU.10 Conocer estado de cuenta
Actor	Act.2 Cliente (C)
Descripción	El caso de uso permite a Cliente saber información como el estado de la cuenta, la fecha inicial y final de activación
Flujo Básico	<p>El caso de uso INICIA cuando el Cliente pasa la etiqueta por el lector NFC.</p> <ol style="list-style-type: none"> 1. El Sistema adquiere el ID de la etiqueta. 2. GUI busca en la BD el C asignado a esa ID. 3. GUI muestra el perfil del C encontrado y envía información de la cuenta al LCD. 4. C lee la información. <p>FIN del caso de uso.</p>
Flujos Alternos	<p>1. ID no ha sido asignada a ningún Cliente</p> <p>En el paso 2 no se encuentra ningún Cliente con la ID leída. GUI informa al R que la ID no ha sido asignada. El Display LCD muestra al C que no ha sido encontrado. El caso de uso finaliza.</p>
Pre-condiciones	1. Las etiquetas deben ser NFC.
Post-condiciones	1. El Sistema cambia de estado "a la espera de una nueva etiqueta".
Punto de extensión	

Cuadro A.11: Especificación del caso de uso Conocer estado de cuenta

A.3. Casos de uso para el Instructor

Caso de Uso	CU.11 Administrar Ejercicio
Actor	Act.3 Instructor (I)
Descripción	El caso de uso permite al Instructor crear, editar y eliminar los Ejercicios.
Flujo Básico	El caso de uso INICIA cuando el Instructor oprime el botón ‘Ejercicios’ desde su interfaz gráfica. 1. GUI muestra las opciones que puede realizar el I en la pantalla. 2. I Elige una opción en la interfaz gráfica. FIN del caso de uso.
Flujos Alternos	
Pre-condiciones	1. El U debe estar logueado con una cuenta de Instructor.
Post-condiciones	
Punto de extensión	2.1. I Elige Crear Ejercicio, hacer «extend»CU.13 Crear Ejercicio 2.2. I Elige Editar Ejercicio, hacer «extend»CU.14 Editar Ejercicio 2.3. I Elige Eliminar Ejercicio, hacer «extend»CU.15 Eliminar Ejercicio 2.4. I Elige Importar Ejercicios, hacer «extend»CU.16 Importar Ejercicios 2.5. I Elige Exportar Ejercicios, hacer «extend»CU.17 Exportar Ejercicios

Cuadro A.12: Especificación del caso de uso Administrar Ejercicio

A.3. CASOS DE USO PARA EL INSTRUCTOR

Caso de Uso	CU.12 Crear Ejercicio
Actor	Act.3 Instructor (I)
Descripción	El caso de uso permite al Instructor crear Ejercicios para las rutinas de entrenamiento
Flujo Básico	<p>El caso de uso INICIA cuando el Instructor oprime el botón ‘Crear Ejercicio’ desde la ventana ‘Ejercicios’.</p> <ol style="list-style-type: none"> 1. GUI abre la ventana de asistente de creación de ejercicios. 2. I llena los datos de el nuevo ejercicio. 3. I oprime el botón ‘Crear’. 4. GUI almacena los datos de el nuevo ejercicio en la BD y cierra la ventana. <p>FIN del caso de uso.</p>
Flujos Alternos	<ol style="list-style-type: none"> 1. Campos relevantes no fueron completados <p>En el paso 3 el Instructor oprime ‘Crear’ sin completar algunos campos. GUI informa al I que algunos campos relevantes no fueron completados. GUI muestra que campos debe llenar. Se repiten las acciones descritas a partir del paso 2.</p>
Pre-condiciones	<ol style="list-style-type: none"> 1. El U debe estar logueado con una cuenta de Instructor.
Post-condiciones	<ol style="list-style-type: none"> 1. GUI informa al I que el ejercicio ha sido creado. 2. GUI retorna a la ventana principal.
Punto de extensión	

Cuadro A.13: Especificación del caso de uso Crear Ejercicio

A.3. CASOS DE USO PARA EL INSTRUCTOR

Caso de Uso	CU.13 Editar Ejercicio
Actor	Act.3 Instructor (I)
Descripción	El caso de uso permite al Instructor editar Ejercicios para las rutinas de entrenamiento
Flujo Básico	<p>El caso de uso INICIA cuando el Instructor oprime el botón ‘Editar Ejercicio’ desde la ventana ‘Ejercicios’.</p> <ol style="list-style-type: none"> 1. GUI abre la ventana de asistente de edición de ejercicios. 2. I escribe el código de el ejercicio que desea editar. 3. GUI carga los datos actuales de el ejercicio especificado. 4. I Edita los valores y oprime el botón ‘Guardar’. 5. GUI almacena los datos de el ejercicio en la BD y cierra la ventana. <p>FIN del caso de uso.</p>
Flujos Alternos	<ol style="list-style-type: none"> 1. Ejercicio no ha sido encontrado <p>En el paso 3 la GUI no encuentra el Ejercicio con el código especificado por el Instructor. GUI informa al I que el ejercicio no existe. Se repiten las acciones descritas a partir del paso 2.</p>
Pre-condiciones	<ol style="list-style-type: none"> 1. El U debe estar logueado con una cuenta de Instructor.
Post-condiciones	<ol style="list-style-type: none"> 1. GUI informa al I que el ejercicio ha sido editado. 2. GUI retorna a la ventana principal.
Punto de extensión	

Cuadro A.14: Especificación del caso de uso Editar Ejercicio

A.3. CASOS DE USO PARA EL INSTRUCTOR

Caso de Uso	CU.14 Eliminar Ejercicio
Actor	Act.3 Instructor (I)
Descripción	El caso de uso permite al Instructor eliminar Ejercicios para las rutinas de entrenamiento.
Flujo Básico	<p>El caso de uso INICIA cuando el Instructor oprime el botón ‘Eliminar Ejercicio’ desde la ventana ‘Ejercicios’.</p> <ol style="list-style-type: none"> 1. GUI abre la ventana de asistente de eliminación de ejercicios. 2. I selecciona el ejercicio que desea eliminar. 3. I oprime el botón ‘Eliminar’. 4. GUI elimina los datos de el ejercicio seleccionado en la BD y cierra la ventana. <p>FIN del caso de uso.</p>
Flujos Alternos	<p>1. Ningún ejercicio ha sido seleccionado</p> <p>En el paso 3 el Instructor oprime ‘Eliminar’ sin seleccionar el Ejercicio.</p> <p>GUI informa al I que ningún ejercicio fue seleccionado.</p> <p>Se repiten las acciones descritas a partir del paso 2.</p>
Pre-condiciones	1. El U debe estar logueado con una cuenta de Instructor.
Post-condiciones	<ol style="list-style-type: none"> 1. GUI informa al I que el ejercicio ha sido eliminado. 2. GUI retorna a la ventana principal.
Punto de extensión	

Cuadro A.15: Especificación del caso de uso Eliminar Ejercicio

Caso de Uso	CU.15 Importar Ejercicios
Actor	Act.3 Instructor (I)
Descripción	El caso de uso permite al Instructor importar Ejercicios de rutinas de entrenamiento a la Base de Datos.
Flujo Básico	El caso de uso INICIA cuando el Instructor oprime el botón ‘Importar Ejercicios’ desde la ventana ‘Ejercicios’. 1. GUI abre la ventana de asistente de importación de ejercicios. 2. I especifica la ruta de ubicación del archivo (.txt) con los ejercicios a importar. 3. I oprime el botón ‘Importar’. 4. GUI reemplaza los ejercicios almacenados en la BD por los nuevos ejercicios del archivo (.txt) y cierra la ventana. FIN del caso de uso.
Flujos Alternos	1. No se especifico la ruta de ubicación En el paso 3 el Instructor oprime ‘Importar’ sin colocar la ruta donde se encuentran los ejercicios. GUI informa al I que no se especifico la ruta de ubicación del archivo (.txt). Se repiten las acciones descritas a partir del paso 2.
Pre-condiciones	1. El U debe estar logueado con una cuenta de Instructor.
Post-condiciones	1. GUI informa al I que la importación de ejercicios ha finalizado con éxito. 2. GUI retorna a la ventana principal.
Punto de extensión	

Cuadro A.16: Especificación del caso de uso Importar Ejercicios

A.3. CASOS DE USO PARA EL INSTRUCTOR

Caso de Uso	CU.16 Exportar Ejercicios
Actor	Act.3 Instructor (I)
Descripción	El caso de uso permite al Instructor exportar Ejercicios de rutinas de entrenamiento a la Base de Datos.
Flujo Básico	<p>El caso de uso INICIA cuando el Instructor oprime el botón ‘Exportar Ejercicios’ desde la ventana ‘Ejercicios’.</p> <ol style="list-style-type: none"> 1. GUI abre la ventana de asistente de exportación de ejercicios. 2. I especifica la ruta de ubicación donde quiere que se guarde el archivo (.txt) con los ejercicios a exportar. 3. I oprime el botón ‘Exportar’. 4. GUI guarda los ejercicios almacenados en la BD en un archivo (.txt) en la ruta especificada y cierra la ventana. <p>FIN del caso de uso.</p>
Flujos Alternos	<ol style="list-style-type: none"> 1. No se especifico la ruta de ubicación <p>En el paso 3 el Instructor oprime ‘Exportar’ sin especificar la ruta de ubicación.</p> <p>GUI informa al I que no se especifico la ruta de ubicación para el archivo (.txt).</p> <p>Se repiten las acciones descritas a partir del paso 2.</p> <ol style="list-style-type: none"> 2. Reemplazar archivo existente <p>En el paso 3 el Instructor oprime ‘Exportar’ pero ya existe en esa dirección un archivo (.txt) con el mismo nombre.</p> <p>GUI pregunta al I si desea reemplazarlo.</p> <p>I selecciona a el botón ‘Si’.</p> <p>GUI borra el archivo (.txt) existente.</p> <p>Se repiten las acciones descritas a partir del paso 4.</p>
Pre-condiciones	<ol style="list-style-type: none"> 1. El U debe estar logueado con una cuenta de Instructor.
Post-condiciones	<ol style="list-style-type: none"> 1. GUI informa al I que la exportación de ejercicios ha finalizado con éxito. 2. GUI retorna a la ventana principal.
Punto de extensión	

Cuadro A.17: Especificación del caso de uso Exportar Ejercicios

Caso de Uso	CU.17 Administrar Rutina
Actor	Act.3 Instructor (I)
Descripción	El caso de uso permite al Instructor crear, editar y eliminar las Rutinas de entrenamiento.
Flujo Básico	El caso de uso INICIA cuando el Instructor oprime el botón ‘Rutinas’ desde su interfaz gráfica. 1. GUI muestra las opciones que puede realizar el I en la pantalla. 2. I Elige una opción en la interfaz gráfica. FIN del caso de uso.
Flujos Alternos	
Pre-condiciones	1. El U debe estar logueado con una cuenta de Instructor.
Post-condiciones	
Punto de extensión	2.1. I Elige Crear Rutina, hacer «extend»CU.19 Crear Rutina 2.2. I Elige Editar Rutina, hacer «extend»CU.20 Editar Rutina 2.3. I Elige Eliminar Rutina, hacer «extend»CU.21 Eliminar Rutina 2.4. I Elige Generar PDF de rutinas, hacer «extend»CU.22 Generar PDF de rutinas

Cuadro A.18: Especificación del caso de uso Administrar Rutina

A.3. CASOS DE USO PARA EL INSTRUCTOR

Caso de Uso	CU.18 Crear Rutina
Actor	Act.3 Instructor (I)
Descripción	El caso de uso permite al Instructor crear Rutinas de entrenamiento
Flujo Básico	<p>El caso de uso INICIA cuando el Instructor oprime el botón ‘Crear Rutina’ desde la ventana ‘Rutinas’.</p> <ol style="list-style-type: none"> 1. GUI abre la ventana de asistente de creación de rutinas. 2. I llena los datos de la nueva rutina. 3. I oprime el botón ‘Crear’. 4. GUI almacena los datos de la nueva rutina en la BD y cierra la ventana. <p>FIN del caso de uso.</p>
Flujos Alternos	<ol style="list-style-type: none"> 1. Campos relevantes no fueron completados <p>En el paso 3 el Instructor oprime ‘Crear’ sin completar algunos campos. GUI informa al I que algunos campos relevantes no fueron completados. GUI muestra que campos debe llenar. Se repiten las acciones descritas a partir del paso 2.</p>
Pre-condiciones	<ol style="list-style-type: none"> 1. El U debe estar logueado con una cuenta de Instructor.
Post-condiciones	<ol style="list-style-type: none"> 1. GUI informa al I que la rutina ha sido creada. 2. GUI retorna a la ventana principal.
Punto de extensión	

Cuadro A.19: Especificación del caso de uso Crear Rutina

Caso de Uso	CU.19 Editar Rutina
Actor	Act.3 Instructor (I)
Descripción	El caso de uso permite al Instructor editar Rutinas de entrenamiento
Flujo Básico	<p>El caso de uso INICIA cuando el Instructor oprime el botón ‘Editar Rutina’ desde la ventana ‘Rutinas’.</p> <ol style="list-style-type: none"> 1. GUI abre la ventana de asistente de edición de rutinas. 2. I escribe el código de la rutina que desea editar. 3. GUI carga los datos actuales de la rutina especificada. 4. I Edita los valores y oprime el botón ‘Guardar’. 5. GUI almacena los datos de la rutina en la BD y cierra la ventana. <p>FIN del caso de uso.</p>
Flujos Alternos	<ol style="list-style-type: none"> 1. Rutina no ha sido encontrada <p>En el paso 3 la GUI no encuentra la Rutina con el código especificado por el Instructor. GUI informa al I que la rutina no existe. Se repiten las acciones descritas a partir del paso 2.</p>
Pre-condiciones	<ol style="list-style-type: none"> 1. El U debe estar logueado con una cuenta de Instructor.
Post-condiciones	<ol style="list-style-type: none"> 1. GUI informa al I que la rutina ha sido editada. 2. GUI retorna a la ventana principal.
Punto de extensión	

Cuadro A.20: Especificación del caso de uso Editar Rutina

A.3. CASOS DE USO PARA EL INSTRUCTOR

Caso de Uso	CU.20 Eliminar Rutina
Actor	Act.3 Instructor (I)
Descripción	El caso de uso permite al Instructor eliminar Rutinas de entrenamiento.
Flujo Básico	<p>El caso de uso INICIA cuando el Instructor oprime el botón ‘Eliminar Rutina’ desde la ventana ‘Rutinas’.</p> <ol style="list-style-type: none"> 1. GUI abre la ventana de asistente de eliminación de rutinas. 2. I selecciona la rutina que desea eliminar. 3. I oprime el botón ‘Eliminar’. 4. GUI elimina los datos de la rutina seleccionada en la BD y cierra la ventana. <p>FIN del caso de uso.</p>
Flujos Alternos	<ol style="list-style-type: none"> 1. Ninguna rutina ha sido seleccionada <p>En el paso 3 el Instructor oprime ‘Eliminar’ sin seleccionar la rutina.</p> <p>GUI informa al I que ninguna rutina fue seleccionada.</p> <p>Se repiten las acciones descritas a partir del paso 2.</p>
Pre-condiciones	<ol style="list-style-type: none"> 1. El U debe estar logueado con una cuenta de Instructor.
Post-condiciones	<ol style="list-style-type: none"> 1. GUI informa al I que la rutina ha sido eliminada. 2. GUI retorna a la ventana principal.
Punto de extensión	

Cuadro A.21: Especificación del caso de uso Eliminar Rutina

Caso de Uso	CU.21 Generar PDF de rutinas
Actor	Act.3 Instructor (I)
Descripción	El caso de uso permite al Instructor generar Rutinas de entrenamiento de cada Cliente en formato PDF.
Flujo Básico	<p>El caso de uso INICIA cuando el Instructor oprime el botón ‘Ver Listado’ desde la ventana ‘Rutinas’.</p> <ol style="list-style-type: none"> 1. GUI muestra en la pantalla todas las rutinas de entrenamiento organizadas. 2. I busca la rutina que desea generar y la abre. 3. GUI muestra la rutina en una nueva ventana organizada por días. 4. I presiona el botón ‘Generar PDF’. 5. GUI crea un archivo de formato PDF organizado por días. <p>FIN del caso de uso.</p>
Flujos Alternos	
Pre-condiciones	<ol style="list-style-type: none"> 1. El U debe estar logueado con una cuenta de Instructor. 2. La Rutina ya debe estar creada.
Post-condiciones	<ol style="list-style-type: none"> 1. GUI abre el archivo PDF creado.
Punto de extensión	

Cuadro A.22: Especificación del caso de uso Generar PDF de rutinas

A.4. Casos de uso para el Recepcionista

Caso de Uso	CU.22 Administrar Cliente
Actor	Act.4 Recepcionista (R)
Descripción	El caso de uso permite al Recepcionista crear, editar y eliminar cuentas a los Clientes del gimnasio.
Flujo Básico	El caso de uso INICIA cuando el Recepcionista oprime el Botón ‘Clientes’ desde su interfaz gráfica. 1. GUI muestra las opciones que puede realizar el R en la pantalla. 2. R Elige una opción en la interfaz gráfica. FIN del caso de uso.
Flujos Alternos	
Pre-condiciones	1. El U debe estar logueado con una cuenta de Recepcionista.
Post-condiciones	
Punto de extensión	2.1. R Elige Crear Cliente, hacer «extend»CU.24 Crear Cliente 2.2. R Elige Editar Cliente, hacer «extend»CU.25 Editar Cliente 2.3. R Elige Eliminar Cliente, hacer «extend»CU.26 Eliminar Cliente 2.4. R Elige Asignar Identificación, hacer «extend»CU.27 Asignar Identificaciones

Cuadro A.23: Especificación del caso de uso Administrar Cliente

A.4. CASOS DE USO PARA EL RECEPCIONISTA

Caso de Uso	CU.23 Crear Cliente
Actor	Act.4 Recepcionista (R)
Descripción	El caso de uso permite al Recepcionista crear cuentas de Clientes.
Flujo Básico	<p>El caso de uso INICIA cuando el Recepcionista oprime el botón ‘Crear Cliente’ desde la ventana ‘Clientes’.</p> <ol style="list-style-type: none"> 1. GUI abre la ventana de asistente de creación de clientes. 2. R llena los datos de el nuevo C. 3. R oprime el botón ‘Crear’. 4. GUI almacena los datos de el nuevo C en la BD y cierra la ventana. <p>FIN del caso de uso.</p>
Flujos Alternos	<ol style="list-style-type: none"> 1. Campos relevantes no fueron completados <p>En el paso 3 el Recepcionista oprime ‘Crear’ sin completar algunos campos. GUI informa al R que algunos campos relevantes no fueron completados. GUI muestra que campos debe llenar. Se repiten las acciones descritas a partir del paso 2.</p>
Pre-condiciones	<ol style="list-style-type: none"> 1. El U debe estar logueado con una cuenta de Recepcionista.
Post-condiciones	<ol style="list-style-type: none"> 1. GUI informa al R que el C ha sido creado. 2. GUI retorna a la ventana principal.
Punto de extensión	2.1. R Selecciona Tomar Foto, hacer «extend»CU.28 Tomar Fotografías

Cuadro A.24: Especificación del caso de uso Crear Cliente

A.4. CASOS DE USO PARA EL RECEPCIONISTA

Caso de Uso	CU.24 Editar Cliente
Actor	Act.4 Recepcionista (R)
Descripción	El caso de uso permite al Recepcionista editar cuentas de Clientes.
Flujo Básico	<p>El caso de uso INICIA cuando el Recepcionista oprime el botón ‘Editar Cliente’ desde la ventana ‘Clientes’.</p> <ol style="list-style-type: none"> 1. GUI abre la ventana de asistente de edición de clientes. 2. R escribe el DNI de el C que desea editar. 3. GUI carga los datos actuales de el C especificado. 4. R Edita los valores y oprime el botón ‘Guardar’. 5. GUI almacena los datos de el C en la BD y cierra la ventana. <p>FIN del caso de uso.</p>
Flujos Alternos	<ol style="list-style-type: none"> 1. Cliente no ha sido encontrado <p>En el paso 3 la GUI no encuentra el Cliente con el DNI especificado por el Recepcionista. GUI informa al R que el C no existe. Se repiten las acciones descritas a partir del paso 2.</p>
Pre-condiciones	<ol style="list-style-type: none"> 1. El U debe estar logueado con una cuenta de Recepcionista.
Post-condiciones	<ol style="list-style-type: none"> 1. GUI informa al R que el C ha sido editado. 2. GUI retorna a la ventana principal.
Punto de extensión	4.1. R Selecciona Editar Foto, hacer «extend»CU.28 Tomar Fotografías

Cuadro A.25: Especificación del caso de uso Editar Cliente

A.4. CASOS DE USO PARA EL RECEPCIONISTA

Caso de Uso	CU.25 Eliminar Cliente
Actor	Act.4 Recepcionista (R)
Descripción	El caso de uso permite al Recepcionista eliminar cuentas de Clientes.
Flujo Básico	<p>El caso de uso INICIA cuando el Recepcionista oprime el botón ‘Eliminar Cliente’ desde la ventana ‘Clientes’.</p> <ol style="list-style-type: none"> 1. GUI abre la ventana de asistente de eliminación de clientes. 2. R escribe el DNI de el C que desea eliminar. 3. GUI muestra información de el C descrito por el DNI. 4. R oprime el botón ‘Eliminar’. 5. GUI elimina los datos de el C seleccionado en la BD y cierra la ventana. <p>FIN del caso de uso.</p>
Flujos Alternos	<ol style="list-style-type: none"> 1. Cliente no ha sido encontrado <p>En el paso 3 la GUI no encuentra el Cliente con el documento especificado por el Recepcionista. GUI informa al R que el C no existe. Se repiten las acciones descritas a partir del paso 2.</p>
Pre-condiciones	<ol style="list-style-type: none"> 1. El usuario del sistema debe estar logueado con una cuenta de Recepcionista.
Post-condiciones	<ol style="list-style-type: none"> 1. GUI informa al R que el C ha sido eliminado. 2. GUI retorna a la ventana principal.
Punto de extensión	

Cuadro A.26: Especificación del caso de uso Eliminar Cliente

A.4. CASOS DE USO PARA EL RECEPCIONISTA

Caso de Uso	CU.26 Asignar Identificaciones
Actor	Act.4 Recepcionista (R)
Descripción	El caso de uso permite al Recepcionista asignar una ID de etiqueta a un Cliente.
Flujo Básico	<p>El caso de uso INICIA cuando un Cliente quiere activar su cuenta.</p> <ol style="list-style-type: none"> 1. R pide el numero de DNI del C. 2. R busca al C con el numero de DNI en la BD. 3. R abre el perfil del C encontrado. 4. R verifica el pago del C. 5. R entrega una etiqueta al C. 6. R presiona el botón ‘Agregar ID’ en el perfil del C. 7. C pasa la etiqueta por el lector de NFC. 8. El Sistema adquiere la ID de la etiqueta. 9. GUI asigna esa ID al C. <p>FIN del caso de uso.</p>
Flujos Alternos	<ol style="list-style-type: none"> 1. No hay pagos de activación <p>En el paso 4 no se encuentra ningún pago activo o disponible del Cliente. GUI muestra al R que no hay pagos disponibles o activos. R informa a C. El caso de uso finaliza.</p>
Pre-condiciones	<ol style="list-style-type: none"> 1. El U debe estar logueado con una cuenta de Recepcionista. 2. Las etiquetas deben ser NFC. 3. La cuenta del C ya debe haberse creado.
Post-condiciones	<ol style="list-style-type: none"> 1. GUI informa al R que la ID ha sido asignada al C.
Punto de extensión	

Cuadro A.27: Especificación del caso de uso Asignar identificaciones

Caso de Uso	CU.27 Tomar Fotografías
Actor	Act.4 Recepcionista (R)
Descripción	El caso de uso permite al Recepcionista tomar fotografías a los Clientes.
Flujo Básico	<p>El caso de uso INICIA cuando el Recepcionista oprime el botón ‘Tomar Foto’ en la ventana de ‘Crear Cliente’ o ‘Editar Foto’ en la ventana de ‘Editar Cliente’.</p> <ol style="list-style-type: none"> 1. GUI abre una ventana asistente de captura fotográfica. 2. R pide al C colocarse en frente de la cámara. 3. R oprime el botón ‘Capturar’. 4. GUI muestra la foto y la guarda en un archivo temporal. 5. R oprime el botón ‘Seleccionar’ y cierra la ventana de asistente 6. GUI carga la imagen JPG en la ventana anterior (‘Crear Cliente’ o ‘Editar Cliente’). <p>FIN del caso de uso.</p>
Flujos Alternos	<ol style="list-style-type: none"> 1. La fotografía quedo mal tomada <p>En el paso 4 el Recepcionista observa que foto quedo mal tomada.</p> <p>R informa al C del error.</p> <p>R pide al C capturar una nueva imagen.</p> <p>El caso de uso se repite a partir del paso 2.</p>
Pre-condiciones	<ol style="list-style-type: none"> 1. El U debe estar logueado con una cuenta de Recepcionista.
Post-condiciones	
Punto de extensión	

Cuadro A.28: Especificación del caso de uso Tomar Fotografías

A.4. CASOS DE USO PARA EL RECEPCIONISTA

Caso de Uso	CU.28 Administrar Pago
Actor	Act.4 Recepcionista (R)
Descripción	El caso de uso permite al Recepcionista crear y eliminar Pagos a los Clientes del gimnasio.
Flujo Básico	El caso de uso INICIA cuando el Recepcionista oprime el botón 'Pagos' desde su interfaz gráfica. 1. GUI muestra las opciones que puede realizar el R en la pantalla. 2. R Elige una opción en la interfaz gráfica. FIN del caso de uso.
Flujos Alternos	
Pre-condiciones	1. El U debe estar logueado con una cuenta de Recepcionista.
Post-condiciones	
Punto de extensión	2.1. R Elige Crear Pago, hacer «extend»CU.30 Crear Pago 2.2. R Elige Eliminar Pago, hacer «extend»CU.31 Eliminar Pago 2.3. R Elige Pagar Saldo, hacer «extend»CU.32 Pagar Saldo

Cuadro A.29: Especificación del caso de uso Administrar Pago

A.4. CASOS DE USO PARA EL RECEPCIONISTA

Caso de Uso	CU.29 Crear Pago
Actor	Act.4 Recepcionista (R)
Descripción	El caso de uso permite al Recepcionista crear Pagos a los Clientes.
Flujo Básico	<p>El caso de uso INICIA cuando el Recepcionista oprime el botón ‘Crear Pago’ desde la ventana ‘Pagos’.</p> <ol style="list-style-type: none"> 1. GUI abre la ventana de asistente de creación de pagos. 2. R escribe el DNI de el C al que quiere agregar el pago. 3. GUI busca el C en la BD y carga su nombre. 4. R llena los datos de el nuevo pago y oprime el botón ‘Crear’. 5. GUI almacena los datos del pago en la BD y cierra la ventana. <p>FIN del caso de uso.</p>
Flujos Alternos	<p>1. Cliente no ha sido encontrado En el paso 3 la GUI no encuentra el Cliente con el DNI especificado por el Recepcionista. GUI informa al R que el C no existe. Se repiten las acciones descritas a partir del paso 2.</p> <p>2. Campos relevantes no fueron completados En el paso 4 el Recepcionista oprime ‘Crear’ sin completar algunos campos. GUI informa al R que algunos campos relevantes no fueron completados. GUI muestra que campos debe llenar. Se repiten las acciones descritas a partir del paso 2.</p>
Pre-condiciones	1. El U debe estar logueado con una cuenta de Recepcionista.
Post-condiciones	<ol style="list-style-type: none"> 1. GUI informa al R que el Pago ha sido creado. 2. GUI retorna a la ventana principal.
Punto de extensión	

Cuadro A.30: Especificación del caso de uso Crear Pago

A.4. CASOS DE USO PARA EL RECEPCIONISTA

Caso de Uso	CU.30 Eliminar Pago
Actor	Act.4 Recepcionista (R)
Descripción	El caso de uso permite al Recepcionista eliminar Pagos a los Clientes.
Flujo Básico	<p>El caso de uso INICIA cuando el Recepcionista oprime el botón ‘Eliminar Pago’ desde la ventana ‘Pagos’.</p> <ol style="list-style-type: none"> 1. GUI abre la ventana de asistente de eliminación de Pagos. 2. R escribe el DNI de el C al que desea eliminar el pago. 3. GUI busca el C en la BD y carga su nombre. 4. GUI muestra los pagos de el C especificado. 5. R selecciona el pago que desea eliminar. 6. R oprime el botón ‘Eliminar’. <p>5. GUI elimina los datos de el pago seleccionado en la BD y cierra la ventana. FIN del caso de uso.</p>
Flujos Alternos	<p>1. Cliente no ha sido encontrado En el paso 3 la GUI no encuentra el Cliente con el DNI especificado por el Recepcionista. GUI informa al R que el C no existe. Se repiten las acciones descritas a partir del paso 2.</p> <p>2. El Cliente no tiene ningún Pago En el paso 4 la GUI no encuentra ningún pago para el Cliente con el DNI especificado por el Recepcionista. GUI muestra al R que el C no ha realizado pagos. Fin de caso de uso.</p> <p>3. Ningún pago ha sido seleccionado En el paso 6 el Recepcionista oprime ‘Eliminar’ sin seleccionar el pago. GUI informa a el R que ningún pago fue seleccionado. Se repiten las acciones descritas a partir del paso 5.</p>
Pre-condiciones	1. El U debe estar logueado con una cuenta de Recepcionista.
Post-condiciones	<ol style="list-style-type: none"> 1. GUI informa al R que el Pago ha sido eliminado. 2. GUI retorna a la ventana principal.
Punto de extensión	

Cuadro A.31: Especificación del caso de uso Eliminar Pago

A.4. CASOS DE USO PARA EL RECEPCIONISTA

Caso de Uso	CU.31 Pagar Saldo
Actor	Act.4 Recepcionista (R)
Descripción	El caso de uso permite al Recepcionista pagar los saldos pendientes a los Clientes.
Flujo Básico	<p>El caso de uso INICIA cuando el Recepcionista oprime el botón ‘Pagar Saldo’ desde la ventana ‘Pagos’.</p> <ol style="list-style-type: none"> 1. GUI abre la ventana de asistente de saldos pendientes. 2. R escribe el DNI de el C del que desea pagar el saldo. 3. GUI busca el C en la BD y carga su nombre. 4. GUI muestra los pagos con saldos de el C especificado. 5. R selecciona el pago que desea pagar el saldo. 6. R escribe la cantidad que el C desea saldar. 7. R oprime el botón ‘Pagar’. 8. GUI actualiza los datos del pago seleccionado en la BD y cierra la ventana. <p>FIN del caso de uso.</p>
Flujos Alternos	<ol style="list-style-type: none"> 1. Cliente no ha sido encontrado En el paso 3 la GUI no encuentra el Cliente con el DNI especificado por el Recepcionista. GUI informa al R que el C no existe. Se repiten las acciones descritas a partir del paso 2. 2. El Cliente no tiene ningún Pago con saldos En el paso 4 la GUI no encuentra ningún pago para el Cliente con el DNI especificado por el Recepcionista. GUI muestra al R que el C no tiene saldos pendientes. R informa al C. Fin de caso de uso. 3. No se escribió la cantidad que desea saldar En el paso 7 el Recepcionista oprime ‘Pagar’ sin haber escrito la cantidad que se desea saldar. GUI informa a el R del error. Se repiten las acciones descritas a partir del paso 6.
Pre-condiciones	1. El U debe estar logueado con una cuenta de Recepcionista.
Post-condiciones	<ol style="list-style-type: none"> 1. GUI informa al R que el Pago ha sido saldado. 2. GUI retorna a la ventana principal.
Punto de extensión	

Cuadro A.32: Especificación del caso de uso Pagar Saldo

A.4. CASOS DE USO PARA EL RECEPCIONISTA

Caso de Uso	CU.9 Identificar
Actor	Act.2 Cliente (C), Act.4 Recepcionista (R)
Descripción	El caso de uso permite la identificación del Cliente.
Flujo Básico	<p>El caso de uso INICIA cuando el Cliente pasa la etiqueta por el lector NFC.</p> <ol style="list-style-type: none"> 1. El Sistema adquiere el ID de la etiqueta. 2. GUI busca en la BD el C asignado a esa ID. 3. GUI muestra el perfil del C encontrado. 4. R verifica que el estado del C este activo. 5. R compara la foto del perfil con el C. 6. R permite el acceso al C. <p>FIN del caso de uso.</p>
Flujos Alternos	<ol style="list-style-type: none"> 1. ID no ha sido asignada a ningún Cliente En el paso 2 no se encuentra ningún Cliente con la ID leída. GUI informa al R que la ID no ha sido asignada. El LCD muestra al C que no ha sido encontrado. El caso de uso finaliza. 2. El estado del Cliente no esta activo En el paso 4 el estado del Cliente no esta activo. R advierte al C que su estado no esta activo y no permite su acceso. El caso de uso finaliza. 3. La foto del perfil no coincide con el Cliente En el paso 5 la foto del perfil no coincide con la imagen del Cliente. R pide el DNI del C. R compara datos importantes como numero del DNI, nombre completo. Si los datos coinciden, hacer CU.28 Tomar Fotografías y R permite el acceso al C.
Pre-condiciones	1. Las etiquetas deben ser NFC.
Post-condiciones	1. El Sistema cambia de estado "a la espera de una nueva etiqueta".
Punto de extensión	

Cuadro A.33: Especificación del caso de uso Identificar

B. Código de Arduino

B.1. Declaración de variables.

```
#include <Adafruit_NFCShield_I2C.h>
#include <Wire.h>
#include <LiquidCrystal.h>

#define IRQ    (2)
#define RESET (3)

Adafruit_NFCShield_I2C nfc(IRQ, RESET);
LiquidCrystal lcd(6, 7, 10, 11, 12, 13);

int k = 0;
boolean Led = false;
boolean Conexion = false;
boolean Lectura;
int j, c;

byte Pesas[8] = {
  B11011,
  B11111,
  B11011,
  B00000,
  B00000,
  B11011,
  B11111,
  B11011 };
```

B.2. Preparación setup().

```
void setup(void)
{
  Serial.begin(115200);
  nfc.begin();
  lcd.begin(20, 4);

  uint32_t versiondata = nfc.getFirmwareVersion();
  if (! versiondata)
```

B.3. BUCLE PRINCIPAL LOOP().

```
{
  lcd.home();
  lcd.print("No se encontro el Lector PN532");
  while (1);
}

nfc.SAMConfig();
lcd.createChar(0, Pesas);
pinMode(8, OUTPUT);
pinMode(9, OUTPUT);
}
```

B.3. Bucle principal loop().

```
void loop(void)
{
  while(Conexion == false)
  {
    delay(500);
    if(Serial.available())
    {
      while(Serial.available() > 0)
      {
        if(Serial.read() == '$')
        {
          if(Serial.read() == '@')
          {
            if(Serial.read() == '%')
            {
              Conexion = true;
              lcd.clear();
              lcd.setCursor(6, 1);
              lcd.write("SISTEMA");
              lcd.setCursor(5, 2);
              lcd.write("CONECTADO");
              digitalWrite(8, HIGH);
              delay(400);
              digitalWrite(8, LOW);
              digitalWrite(9, HIGH);
              delay(400);
              digitalWrite(9, LOW);
              digitalWrite(8, HIGH);
              delay(400);
              digitalWrite(8, LOW);
              digitalWrite(9, HIGH);
              delay(400);
              digitalWrite(9, LOW);
            }
          }
        }
      }
    }
  }
}
```

```
        }
    }
}

if(Led == false)
{
    digitalWrite(8, HIGH);
    Led = true;
    if(k == 4)
    {
        lcd.clear();
    }
    else
    {
        k++;
    }
}
else
{
    digitalWrite(8, LOW);
    Led = false;
}

uint8_t uid[] = {0, 0, 0, 0, 0, 0, 0};
uint8_t Longitud;
Lectura = nfc.readPassiveTargetID(PN532_MIFARE_ISO14443A, uid,
    &Longitud, 3000);

if(Lectura)
{
    digitalWrite(8, LOW);
    digitalWrite(9, HIGH);
    String ID;
    for(uint8_t i=0; i<Longitud; i++)
    {
        if (uid[i] <= 0xF)
        {
            ID += "0";
        }
        ID += String(uid[i], HEX);
    }
    Serial.print(ID);
    delay(500);

    if (Serial.available())
    {
        if(char(Serial.peek())=='>')
        {
            Serial.read();
            lcd.clear();
        }
    }
}
```

B.3. BUCLE PRINCIPAL LOOP().

```
    lcd.setCursor(0, 3);
    lcd.write("finalizado con exito");
    lcd.setCursor(3, 2);
    lcd.write("La lectura ha");
    lcd.setCursor(2, 0);
    lcd.write("ETIQUETA LEIDA:");
}
else if(char(Serial.peek())=='<')
{
    Serial.read();
    lcd.clear();
    lcd.setCursor(3, 3);
    lcd.write("en la lectura");
    lcd.setCursor(0, 2);
    lcd.write("Ha ocurrido un error");
    lcd.setCursor(7, 0);
    lcd.write("ERROR:");
}
else
{
    Saludo();
    c = 4;
    String Nombre;
    while (Serial.available() > 0)
    {
        byte Letra = Serial.read();
        if(Letra == '&')
        {
            c--;
            lcd.setCursor(0, c);
            switch(c)
            {
                case 0:
                    j = 0;
                    break;
                case 1:
                    lcd.write("Dias Restantes: ");
                    break;
                case 2:
                    Letra = Serial.read();
                    if(char(Letra)=='A')
                    {
                        lcd.write("F.Inicio: ");
                    }
                    else if(char(Letra)=='P')
                    {
                        lcd.write("Estado: PENDIENTE");
                    }
                    else if(char(Letra)=='I')
                    {
                        lcd.write("Estado: INACTIVO");
                    }
            }
        }
    }
}
```

```
        break;
        case 3:
        Letra = Serial.read();
        if(char(Letra)=='A')
        {
            lcd.write("F.Final: ");
        }
        else if(char(Letra)=='P')
        {
            lcd.write("Iniciara: ");
        }
        else if(char(Letra)=='I')
        {
            lcd.write("Finalizo: ");
        }
        break;
    }
}
else
{
    if(c == 0)
    {
        Nombre += char(Letra);
        if(j < 19)
        {
            lcd.print(char(Letra));
        }
        else if(j >= 19)
        {
            lcd.home();
            lcd.print(Nombre.substring(j-19));
            delay(500);
        }
        j++;
    }
    else
    {
        lcd.print(char(Letra));
    }
}
}
delay(1000);
lcd.home();
lcd.print(Nombre.substring(0, 20));
}
k=0;
}
if(Conexion == true)
{
    digitalWrite(9, LOW);
    digitalWrite(8, HIGH);
    Led = true;
}
```

B.4. FUNCIÓN DE BIENVENIDA SALUDO().

```
    }
  }

  if (Serial.available())
  {
    while (Serial.available() > 0)
    {
      if(Serial.read() == '%')
      {
        if(Serial.read() == '@')
        {
          if(Serial.read() == '$')
          {
            Conexion = false;
            lcd.clear();
            digitalWrite(8, HIGH);
            delay(400);
            digitalWrite(8, LOW);
            digitalWrite(9, HIGH);
            delay(400);
            digitalWrite(9, LOW);
          }
        }
      }
    }
  }
}
```

B.4. Función de bienvenida Saludo().

```
void Saludo()
{
  lcd.clear();
  lcd.setCursor(5, 1);
  lcd.write("BIENVENIDO");
  delay(150);
  lcd.setCursor(6, 3);
  lcd.print(" GYM ID ");
  delay(150);
  lcd.home();
  lcd.print(char(0));
  delay(150);
  lcd.print(char(0));
  delay(150);
  lcd.print(char(0));
  delay(150);
  lcd.print(char(0));
  delay(150);
}
```

```
    lcd.setCursor(3, 1);  
    lcd.print(char(0));  
    delay(150);  
    lcd.setCursor(3, 2);  
    lcd.print(char(0));  
    delay(150);  
    lcd.setCursor(3, 3);  
    lcd.print(char(0));  
    delay(150);  
    lcd.print(char(0));  
    delay(150);  
    lcd.print(char(0));  
    delay(150);  
    lcd.setCursor(14, 3);  
    lcd.print(char(0));  
    delay(150);  
    lcd.print(char(0));  
    delay(150);  
    lcd.print(char(0));  
    delay(150);  
    lcd.setCursor(16, 2);  
    lcd.print(char(0));  
    delay(150);  
    lcd.setCursor(16, 1);  
    lcd.print(char(0));  
    delay(150);  
    lcd.setCursor(16, 0);  
    lcd.print(char(0));  
    delay(150);  
    lcd.print(char(0));  
    delay(150);  
    lcd.print(char(0));  
    delay(150);  
    lcd.print(char(0));  
    delay(150);  
    lcd.clear();  
}
```

C. Diseño de la cubierta del Lector NFC

Para el diseño de la cubierta se toma las medidas de altura, anchura y largo del lector de etiquetas NFC ensamblado, la pantalla LCD y el LED bicolor. La Figura C.1 muestra el diseño con las dimensiones principales de la cubierta del lector de etiquetas NFC en milímetros.

Los detalles en el diseño son muy importantes para el montaje, por este motivo se coloca un marco a la medida donde se pondrá la pantalla LCD y el LED Bicolor para que queden bien posicionados, se disminuye el espesor de la cubierta donde va la antena del lector para disminuir la interferencia de comunicación. De igual manera sobresalen los bordes de la cubierta para que las piezas superior e inferior encajen bien.

Se agregan barras que sobresalen al plano de la pieza inferior para fijar el lector ensamblado por medio de los huecos que trae por defecto el Arduino. También se deja el espacio en la parte trasera para sacar el puerto USB del Arduino, la ubicación de este espacio fue calculado teniendo en cuenta el espesor de la "cama" que se coloca en material aislante de electricidad por debajo del Arduino.

Por último se agregan cajas solidas con orificios que servirán para meter los tornillos, ya sea para atornillar la pieza inferior con la superior de la cubierta o para fijar el LCD.

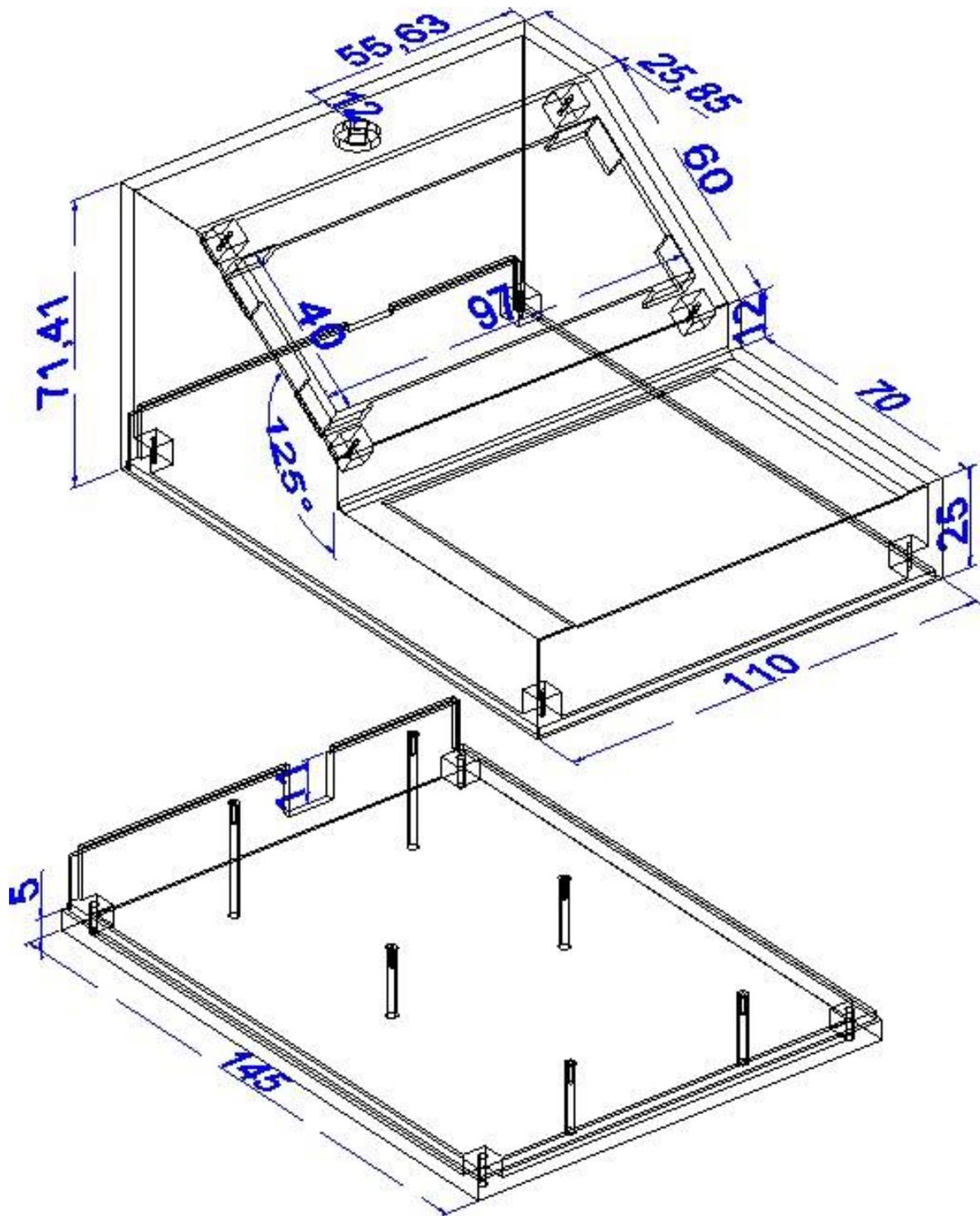


Figura C.1: Dimensiones de la cubierta del Lector de etiquetas NFC