
TECNÓLOGICO DE ESTUDIOS SUPERIORES DE TIANGUISTENCO

MANIPULACIÓN DE HARDWARE BIOMÉTRICO PARA LLEVAR EL CONTROL DE TRABAJADORES

TESIS

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO MECATRÓNICO

PRESENTA:

FELIPE CARRASCO COMONFORT

DIRECTOR (A):

DR. MARIO IBAÑEZ OLVERA

ASESOR(ES):

DR. JUAN NABOR BALDERAS GUTIERREZ

RESUMEN

En el presente proyecto se realizó la programación para tener vinculación con el hardware Biométrico por medio de un conector con la dirección IP y así guardar los datos correspondientes en el Biométrico, a través del programa de C#.

Los datos de un trabajador son nombre, contraseña y huella digital, son datos que se necesitan para tener un guardado exitoso, para guardar los datos que se mencionan se necesitó el uso de un gestor de base de datos con lenguaje SQL, el gestor “MySQL”, ya que es una forma de configurar base de datos para un administrador a un usuario, en este caso el administrador es la persona responsable de registrar los trabajadores nuevos que van contratando en la empresa. El gestor de base de datos es el que usa la empresa “Centro de Validación Digital CVDSA, S.A. de C.V”, para almacenar datos de todos los proyectos que se han realizado a lo largo de los años.

En la base de datos, se realizaron dos tablas, una tabla corresponde al administrador, que es la persona encargada de registrar los nuevos trabajadores, la tabla contiene el acceso para acceder al proyecto, y el acceso que se menciona es el usuario y contraseña de la persona que se encarga de registrar los usuarios nuevos de la empresa, al igual que si son despedidos, el usuario es la encargada de eliminar la persona despedida de la empresa, también se obtienen el menú de los trabajadores registrados, en este caso el acceso que se guarda es el nombre y contraseña, es de muy sería importancia tener seguros estos accesos, porque son datos personales y delicados de cada trabajador.

La segunda tabla que se realizó es la tabla para guardar los nombres de los nuevos trabajadores, o como se menciona en caso de ser despedidos se eliminan de la base de datos, se guarda lo que es un nombre de trabajador y una contraseña, todos los datos que se mencionan, son únicos de cada trabajador de la empresa.

Mediante las ventanas que se realizaron para el registro en C# no fue posible realizar el guardado de la huella digital en la base de datos, únicamente se guardó en el biométrico, como función normal de cualquier biométrico, únicamente se guardó el nombre y contraseña de cada usuario que se fue registrando, en la base de datos y en el biométrico.

Se obtuvo la necesidad de replicar la huella digital a un segundo Biométrico, no se pudo a través de alguna función de C#, se investigó a fondo para saber algún método de réplica de huella digital y se logró obtener el resultado de run_cmd. Run_cmd es un método eficiente para la réplica de la huella digital, mediante clases de otro lenguaje de programación.

Para tener resultados de réplica de huella digital se necesitó del lenguaje de programación de Python, se tuvo que instalar Python para utilizar a través del método run_cmd del lenguaje de programación de C#.

El lenguaje de programación que se tuvo que utilizar fue Python, muy eficiente para los Biométricos, pero únicamente las clases de funciones que contienen código de Python, la función de Run_cmd tiene la función de invocar en el directorio donde se encuentra la clase, así como donde se encuentra instalado el lenguaje de programación de Python.

ABSTRACT

In the present project, programming was carried out to establish a connection with Biometric hardware through a connector with the IP address, enabling the storage of corresponding data in the Biometric system via the C# program.

Worker data includes name, password, and fingerprint – information crucial for successful storage. To save this data, a database management system with SQL language was utilized, specifically "MySQL." This choice provides a means to configure a database for an administrator and a user. In this case, the administrator is responsible for registering new employees in the company.

The selected database management system is used by the company "Centro de Validación Digital CVDSA, S.A. de C.V," to store data from all projects undertaken over the years. Two tables were created in the database. One table corresponds to the administrator, responsible for registering new workers. The table contains access to the project, and the mentioned access includes the username and password of the person responsible for registering new users in the company. Additionally, if individuals are terminated, the administrator is responsible for removing them from the database. The table also retrieves the menu of registered workers, where the access stored includes the name and password. Securing these accesses is of utmost importance due to the personal and sensitive nature of the worker's data.

The second table created is for storing the names of new workers. In case of termination, the data is deleted from the database. This table includes the worker's name and password, unique to each employee.

Through the windows created for registration in C#, it was not possible to save the fingerprint in the database; it was only stored in the biometric device as a standard function. Only the name and password of each registered user were saved in both the database and the biometric device.

There arose a need to replicate the fingerprint to a second biometric device. This couldn't be achieved through C# functions, leading to in-depth research. The solution was found in the "run_cmd" method. "Run_cmd" is an efficient method for fingerprint replication, utilizing classes from another programming language.

To achieve fingerprint replication, the programming language Python was required. Python had to be installed to use the "run_cmd" method in the C# programming language.

The programming language used was Python, which proved to be efficient for biometrics. However, only classes with Python code were utilized. The "Run_cmd" function is designed to invoke in the directory where the class is located, as well as where the Python programming language is installed.

INDICE

Introducción.....	1
PLANTEAMIENTO DEL PROBLEMA	3
Justificación	4
Objetivos.....	5
CAPITULO 1. GENERALIDADES.....	6
1.1 Funcionamiento y rendimiento	8
1.1.1 Reconocimiento de iris /retina.....	9
1.1.2 Reconocimiento Facial.....	10
1.1.3 Autenticación de la voz	11
1.1.4 Dispositivos de captura	11
1.1.5 Sistema de captura	12
1.1.6 Middleware.....	13
1.1.7 Motor biométrico	14
1.2 Programación con Biométricos	14
1.2.1 Python.....	15
1.2.2 C# (C-Sharp)	17
1.2.3 Java.	20
1.2.4 VB. (Visual Basic .NET)	21
1.2.5 Programación C/C++.....	25
1.2.5.1 Curl_global_all().....	26
1.2.5.2 Curl_easy_init().....	27
1.2.5.3 Curl_easy_setopt()	27
1.2.5.3 Curl_easy_cleanup()	27
1.2.5.4 Curl_global_cleanup	27
1.3 Delphi.....	28
1.4 FoxPro.....	29
1.5. Generador de Energía (PowerBuilder)	32
CAPITULO 2. Metodología.	36
2.1. Explicación de uso de Visual Studio 2022	38
2.2. Configuración y programación hardware biométrico	38
2.3. Creación de programación para realizar la interacción del biométrico.	38
2.4. Desarrollo de base de datos y conexión (Interfaz de usuarios).	38
2.5. Réplica de Usuario en dos Biométricos con datos correspondientes.....	39
2.6. Instalación de Python	39
2.7. Replica usuarios resultados con datos correspondientes.....	39
Capítulo 3 Resultados	40
3.1. Explicación de uso de Visual Studio 2022.	40
3.1.1 Configuración y programación hardware biométrico.....	41

3.1.2 BENEFICIOS	41
3.1.3 Asignación de Librería zkemkeeper.....	42
3.1.4 Implementación del Conector de Mysql	43
3.1.5.- Clase utilería para interacción de la dirección IP con el biométrico.....	44
3.1.5.1 La dirección IP	44
3.1.5.2. Direcciones Ip privadas o locales	44
3.1.5.3 Direcciones públicas o remotas.....	45
3.1.5.4 Desarrollo de Login.....	49
3.1.5.6 Form.....	50
3.1.5.7 . Uso de elementos Label (etiqueta) para títulos de datos.....	51
3.1.5.8. Uso de textbox (caja de texto) para introducir y programar datos.	55
4. Desarrollo de base de datos, y conexión (Interfaz de usuarios).	61
4.1 Inicio creación de la base de datos.	61
4.2 CRD	65
4.2.1. Read- Leer - Consultar	65
3.2.2 . Crear y Ejecutar la Consulta SQL.	67
3.2.3. Configuración de datos de base de datos.	70
Número de Registro.....	73
Nombre de trabajador.....	73
Contraseña.....	73
Id Privilegio	73
Agregar Huella	73
3.2.4. Realización de insert con la base de datos y el Biométrico.	73
3.2.5. Verificación de existencia trabajadores en el biométrico.....	82
3.2.6.-Consulta de Usuario en DataGridView de C#.	83
3. 2.7. Código para generar delete en c# y Mysql.....	84
3. 2.8. Código para generar update en c# y Mysql.....	86
4. Replica de usuario en dos biométricos con datos correspondientes.	89
4.1 Instalación de Python y uso de clases de Python.....	95
4,2 . Implementación del Evento Run_Click.....	96
RESULTADOS Y DISCUSIÓN.	100
Conclusiones y recomendaciones.	113
REFERENCIAS	115

ÍNDICE DE FIGURAS

Figura 1. Representación del rendimiento de medida biométrica.	8
Figura 2. Dato Biométrico de la huella digital.	9
Figura 3. Reconocimiento Biométrico por Iris.....	10
Figura 4. Reconocimiento Biométrico facial.	10
Figura 5. Ejemplo de biometría autenticación de la voz.	11
Figura 6. Ejemplo de dispositivo biométrico de captura.	12
Figura 7. Ejemplo de Sistema de captura.....	13
Figura 9. Motor Biométrico	14
Figura 10. Diagrama de bloques metodología para la realización de las simulaciones.	37
Figura 11. Imagen de identificación de Visual Studio 2022	40
Figura 12. Asignación de librería zkemkeeper.	43
Figura 13. Librería zkemkeeper.	43
Figura 14. Archivo MySql.Data.dll	44
Figura 15. Dirección IPv4, la dirección que se está utilizando es 192.168.100.58...	47
Figura 16. Dirección Ip configuración conexión de biométrico.	48
Figura 17. Resultaos de comando ping y dirección Ip en terminal de pc.	48
Figura 18. Nuevo proyecto Aplicación de Windows Forms(NET Framework).	50
Figura 19. Form1 del proyecto Windows Forms NET Framework.	51
Figura 20. Labe1, en español etiqueta elemento para mostrar básicamente un título.	51
Figura 21. Texto de Label asignado el nombre de usuario.	52
Figura 22. Texto de Label asignando el nombre contraseña.	53
Figura 23. Tipo de letra para títulos de datos, Usuario y Contraseña.	54
Figura 24. Despliegue de elemento Font.	54
Figura 25. Opciones para cambiar tamaño de letra para Usuario y Contraseña.....	55
Figura 26. Elemento textbox (caja de texto).	56
Figura 27. Cambio de nombre textbox1 a Nombre.....	56
Figura 28. Cambio de nombre textbox2 txtContra	57
Figura 29. Ejemplo de botones agregados a login.	58
Figura 30. Nombre de botón iniciar.	58
Figura 31. Cambio de botón 1 a iniciar sesión.	59
Figura 32. Nombre de botón Salir.	59
Figura 33. Ejemplo de cambio de título.	60
Figura 34. Asignación de nombre de los botones.	60
Figura 35. Opciones para crear base de datos con Gestor MySQL.	61
Figura 36. Configuración de conexión a la base de datos.....	61
Figura 37. Visualización de la base de datos en blanco.	62

Figura 38. Botón Creación base de datos.....	62
Figura 39. Visualización del nombre de la base de datos.	63
Figura 40. Base de datos correspondientes.	63
Figura 41. Tablas correspondientes.	63
Figura 42. Datos correspondientes de la tabla tc_trabajador.	64
Figura 43. Datos de tabla tt_usuario.	64
Figura 44. Menú de Bienvenido.	72
Figura 45. Nuevo formulario para nuevo método insertar.....	73
Figura 46. TextBox con el nombre de Nombre.	76
Figura 47. TextBox con el nombre de txtContra.	77
Figura 48. Registro de usuario Biométrico.	82
Figura 49 Usuario Registrado en el biométrico.....	83
Figura 50. Dirección IP de Biométrico1.	92
Figura 51. Conexión de Biométrico 2.	92
Figura 53. Configuraciones de la Nueva columna con Botón para Replicar.	93
Tabla 14. Columna Botón de nombre Replicar Usuario.....	94
Figure 54. .exe para instalación lenguaje Python.	95
Figura 55. Instalación satisfactoria de Python.	95
Figura 56. Ubicación de Python.exe.	98
Figura 57 Ubicación de archivos de Python.	99
Figura 58. Clase de Huellas.py.	100
Figura 59 Imagen de clase de voz llamada test_voice.py	103
Figura 60. Usuario agregando nuevo usuario	106
Figura 61. Mario Ibáñez Olvera registrado satisfactoriamente.	107
Figura 62. Usuario registrado en el Biométrico	108
Figura 63. Huellas digitales de Biométrico.	109
Figura 64. Agregando huella digital en Biométrico manera Manual.	110
Figura 65, Imagen de usuario con huella y contraseña.....	110
Figura 66. Consola de lenguaje de programación de Python.	112
Figura 67. Datos satisfactorios de la réplica del Biométrico 2.	112

INDICE DE TABLAS

Tabla 1. Ejemplo de Datagridview, para la base de datos.	69
Tabla 3. Usuario y contraseña de administrador.	70
Tabla 4. Datagridview de consulta de trabajadores.....	72
Tabla 5. Lista de registro de usuarios.....	83
Tabla 6. Consulta del lado de base de datos MySQL.....	84
Tabla 8. Consulta de usuario eliminado.	85
Tabla 9. Usuario eliminado al 100%.....	85
Tabla 10. Select*from de la base de datos para verificar que existe un usuario.....	86
Tabla 11. Verificación de la tabla C# Revisión de usuario existente.	86
Tabla 12. Tabla actualizada MySql.	87
Tabla 13. Verificación de tabla actualizada tabla proyecto.	88
Tabla 15. Nuevo usuario registrado	107
Tabla 16. Base de datos con usuario registrado en la base de datos.	111
Tabla 17. se muestra que se ha presionado el botón de replicar Huella.....	111

Dedicatoria

Esto va dedicado a la persona que me ayudó a terminar este proyecto, Dr Mario Ibáñez Olvera que es la persona que me estuvo orientando para la finalización del proyecto.

Jesús Ángel Jiménez Valdín, Oscar Piña Noverón, Erasto Flores Herrera, Edgar Rodrigo Bueno Matías, Hugo Sebastián Aguilar Nava, personas que me dieron buenas ideas a lo largo de la carrera, mis amigos que considero parte de mi familia Diego Armando Patlán García, Juan Pablo Patlán García de la carrera de Ingeniería Industrial que me daban ánimos para continuar, y la humilde participación de mi amigo de años el ING. Nicolás Serafín Sedano, ya que gracias a su ayuda corregí correcciones del proyecto.

Agradecimientos

A mis amigos Hugo Sebastián Aguilar Nava, Edgar Rodrigo Bueno Matías, José Antonio Martínez Chávez, Emmanuel Juárez Flores, Daniel Peña Serrano, Oscar Piña Noverón, Erasto Ignacio Flores Herrera y de la muy fuerte ayuda de mi buen amigo Jesús ángel Jiménez Valdin, con quienes empezó el camino de mi formación profesional, y más que nada me levantaron cuando estaba completamente abatido, gracias a su ayuda, me encuentro en el final del proceso de mi formación profesional, no me queda más que desearles todo el éxito en su vida profesional “Gracias”.

A la empresa “Centro de Validación Digital CVDSA, S.A. de C.V”, por el apoyo económico otorgado en este proyecto.

GLOSARIO DE ABREVIATURAS

Libcurl: Biblioteca de código abierto ampliamente utilizada para actualizar solicitudes HTTP y gestionar transferencias de datos a través de protocolos.

CURL: Herramienta de línea de comandos.

IDE: Software que proporciona herramientas y funciones integradas para facilitar el desarrollo de software.

IP: Identificador numérico asignado a cada dispositivo conectado a una red de computadoras que utiliza el protocolo de Internet para la comunicación.

Login: Proceso para acceder a un sistema informático u aplicación.

Form: Ventana de Formulario C#-

Label: Control de interfaz gráfica de usuario utilizado para mostrar texto.

Font: Forma de representar información de formato en fuente de texto.

Textbox: Control de interfaz gráfica de usuario para ingresar y editar Texto.

Root: Usuario o administrador con privilegios más elevados en un servidor MySQL.

Datagridview: Control para mostrar y editar datos tabulares en forma de cuadrícula.

Crud: Es un acrónimo que se refiere a las cuatro operaciones básicas en sistemas de gestión de bases de datos.

Sql: Structured Query Language, es un lenguaje de programación diseñado para gestionar y manipular bases de datos relacionales.

Dirección IP: Identificador único asignado a cada dispositivo conectado a una red que utiliza el protocolo de internet para la comunicación.

Biométrico MB160: Es un innovador dispositivo para la gestión de tiempo y asistencia de los empleados, soporta métodos de verificación por medio de huella digital, rostro y contraseña.

Scanner infrarrojo: Dispositivo opto-eléctrico que puede captar y medir la radiación electromagnética infrarroja dentro de un área de visión en específico.

Curl_global_all (): Función para configurar el entorno de programación que lib_curl necesita, se llama al menos una vez dentro de un programa, antes de que el programa llame otra función de libcurl.

Curl_easy_init (): Función que devuelve un identificador CURL, que se debe utilizar como input para otras funciones.

Curl_easy_setopt (): Se utiliza para especificar a libcurl cómo debe comportarse. Al configurar todas las opciones, esta función realizará la función que se comporte de alguna manera.

Curl_easy_cleanup(): Esta función cerrará las conexiones que este identificador ha utilizado y, posiblemente ha mantenido.

Curl_global_cleanup: Esta función libera recursos previamente adquiridos por otra función que debe ser llamada una vez por cada llamada que se realiza.

Zkemkeeper: Librería de ZKTeco exclusiva para Biométricos MB160.

MySql.Data.dll: Archivo para el desarrollo de MySql.Data, exclusivo para el uso del Lenguaje de Programación C#.

Directorio pc: Contenedor virtual en el que se almacenan una agrupación de archivos informáticos y otros subdirectorios.

Protocolo de Internet: Conjunto de reglas para direccionar paquetes de datos para que puedan viajar a través de las redes y llegar al destino correcto.

Router: Dispositivo que proporciona Wi-Fi y que generalmente está conectado a un módem.

Octeto Ip: Cada uno de los conjuntos de números en una dirección IPv4.

Eternet: Es una tecnología estándar de comunicación de redes de área local (LAN, por sus siglas en inglés) que define cómo los dispositivos en una red se comunican entre sí.

Ping computación: Se utiliza para verificar la conectividad entre dispositivos a través de una red, como por ejemplo, entre computadora y un servidor remoto.

Framework: Se refiere a una estructura conceptual y tecnológica que proporciona un conjunto de herramientas, bibliotecas, estándares y directrices que facilitan el desarrollo y la implementación de Software.

Localhost: Se refiere al propio dispositivo en el que se realiza la solicitud, principalmente una red interna.

Introducción

Los datos personales biométricos son características físicas o de comportamiento que pueden ser utilizados para identificar digitalmente a una persona y de esta forma otorgar acceso a sistemas, dispositivos o datos.

Debido a que los datos biométricos pueden proporcionar un nivel razonable de confianza para autenticar a una persona y ser mucho más cómodos para usuarios, esta tecnología tiene el potencial de mejorar la seguridad tanto para las personas como las empresas. Huellas dactilares: El patrón que tienen los dedos es único e irrepetible.

Patrones faciales: Suelen ser únicos también, pues se basan en la combinación de todos los patrones y no solo en un par. Por ello se usan como datos biométricos y si el equipo cuenta con cámara, puede utilizarse para verificar esos patrones.

Retina: De la misma manera que las huellas digitales tienen patrones únicos, las retinas tienen el mismo nivel de exclusividad por persona. Así como las cámaras pueden reconocer los rasgos faciales, las retinas pueden ser escaneadas.

Voz: Es otro de los datos biométricos que se usan para identificar a los usuarios. Actualmente, ya existen asistentes digitales que usan el reconocimiento de voz para identificar a una persona.

ADN: Uno de los datos biométricos por excelencia, aunque es más complicado tener aplicaciones prácticas para los usuarios, ya que se requiere tener una muestra no contaminada del ADN.

Los escáneres biométricos se están volviendo cada vez más sofisticados, por ejemplo, la tecnología de reconocimiento facial de cualquier teléfono inteligente, los infrarrojos sobre el rostro del usuario para realizar una autenticación por coincidencia de patrones. La posibilidad de errar la identificación es una en un millón.

El control de horario se hace cada vez más difícil cuando los sistemas requieren varios medios de identificación, como la detección de gestos naturales (como el parpadeo) y la correspondencia de las codificadas con usuarios dentro de dominios cifrados. Con correspondencia de muestras codificadas con usuarios dentro de dominios cifrados. Con objeto de frustrar los ataques malintencionados, algunos sistemas de seguridad también incluyen en los datos biométricos características adicionales como la edad, el sexo y la altura.

Existen empresas que tienen datos biométricos que funcionan por medio de huella dactilar, el iris de los ojos, las fracciones del rostro, tono de voz, firma, tarjeta electrónica con chip, todo esto sinónimo de datos biométricos, la biometría se auxilia de las matemáticas y de la estadística para emplear los rasgos físicos o conductuales como una forma de identificar la identidad de las personas registradas, todo esto según la fuente. (POLITICO, 2020)

PLANTEAMIENTO DEL PROBLEMA

La empresa Centro de Validación Digital CVDSA, S.A. de C.V área de desarrollo llamada TI y la empresa se encuentra en Toluca Estado de México, no cuenta con un control de horario de manera profesional para tener el control estricto de los trabajadores en cuestión de hora/entrada/comida, por lo cual se pretende en primer lugar verificar los tipos de lenguajes con los cuales es posible manipular el biométrico mb160, y por medio de ello realizar la manipulación del hardware biométrico.

En segundo lugar, verificar las variables que están involucradas en el sistema para que se pueda desarrollar la programación que sea capaz de interactuar con el hardware biométrico.

En tercer lugar, verificar el uso de la base de datos MySQL para tener en desarrollar una base de datos que permita agregar, consultar, actualizar y borrar cualquier usuario registrado en la empresa.

Con este proyecto es posible verificar el control de horario de los trabajadores de la empresa Centro de Validación Digital CVDSA, S.A. de C.V área de TI.

Justificación

La empresa Centro de Validación Digital CVDSA, S.A. de C.V., últimamente ha tenido la problemática de no tener un control de horario con los trabajadores y necesita tener más disciplina con respecto al horario, y así saber que efectivamente han estado cumpliendo las horas laborales.

Ocasionalmente cuando a un trabajador se le solicita presentar algún avance de algún proyecto de cualquier área, normalmente por vía zoom, resulta que no se encuentra el trabajador en el momento que es requerido y es una problemática muy grande porque se necesita con urgencia que el trabajador exponga el avance de cualquier proyecto que se la ha solicitado y de aquí nace la necesidad de tener un control de horario de los trabajadores, de la empresa Centro de Validación Digital CVDSA, S.A. de C.V.

Objetivos

Objetivo General

Programar un hardware biométrico con la ayuda del lenguaje de programación C#, al igual crear la base de datos con los datos de los trabajadores, con el gestor de datos MySQL para el control de entrada/salida/comida del personal de la empresa **Centro de Validación Digital CVDSA, S.A. de C.V.**

Objetivos Específicos.

- 1) Crear una base de datos para tener información y datos de los trabajadores.
- 2) Crear un conector para interactuar con el Hardware Biométrico.
- 3) Desarrollar una interfaz para proporcionar reportes de trabajadores.
- 4) Extraer datos de los trabajadores y guardar en la base de datos.

CAPITULO 1. GENERALIDADES

Durante todo el siglo pasado varias empresas que han concentrado sus esfuerzos en desarrollar sistemas biométricos para garantizar su seguridad, especialmente desarrollando proyectos como los propios departamentos de defensa de varios países.

Encontrar un sistema infalible e inequívoco para reconocer personas es el objetivo último de la biometría. Hoy en día contamos con una gran variedad de equipos capaces de identificar a las personas a partir de la información de alguna parte de su cuerpo como las manos, la retina, el iris, los dedos, las huellas dactilares, la voz o la firma. Incluso se está investigando en la posibilidad de crear un sistema basado en el ADN. Restringiendo a través de su historia por su costo elevado, una función cuestionable y proveedores transitorios, la identificación biométrica está experimentando ahora una aceptación creciente, no solamente en aplicaciones de alta seguridad tales como bancos e instalaciones gubernamentales, sino también en clubes de salud.

De esta forma con los sistemas biométricos que reconocen las características singulares de las huellas digitales, por ejemplo, se logra evitar fraudes en la banca, en el sistema de salud por suplantación de pacientes, controlar el acceso en el desplazamiento de seres humanos al interior de las empresas, tiempos desperdiciados, accesos no deseados, sin necesidad de utilizar contraseñas, tarjetas magnéticas u otros medios de identificación vulnerables. Esto hace que los sistemas biométricos sean el medio más rápido y seguro mediante la utilización de la huella digital como variador de operaciones y de control de acceso. Normalmente para el control de horario las empresas cuentan con un procedimiento biométrico, que es un software que reconoce rasgos físicos determinados, ejemplo, huella dactilar, iris, o la retina, este sistema se utiliza normalmente en todo el mundo, no solamente en las empresas privadas, sino también las empresas de gobierno, todo eso para poder tener identificadas a todas las personas, por medio de un registro, tanto para que las personas registradas obtengan un beneficio por parte de las empresas particulares o de las empresas de gobierno, también para tener un control de acceso o control de horario de trabajadores laborando

en empresas de gobierno y particulares, todo este conocimiento adquirido de la referencia (POLITICO, 2020).

Los biométricos normalmente funcionan con scanner infrarrojo, también se pueden identificar características subcutáneas, por ejemplo, las venas de la mano, la sangre pobre en oxígeno de las venas absorbe más radiación infrarroja que el tejido circundante y se pueden ver con claridad.

Actualmente se están probando métodos de identificación en base al ritmo cardiaco, también la medición de las ondas cerebrales permitirá identificar los datos de los individuos. La identificación con datos biométricos avanza rápido, es muy práctico en la vida diaria, porque el tema de escribir la contraseña de mil caracteres, un número alfanumérico, mayúscula o signo es muy complejo, todos esos datos de contraseña de cualquier carácter se pueden olvidar a la semana siguiente, y con ayuda de cualquier dato biométrico, huella dactilar o algún dato físico de cualquier persona ayuda a que las personas puedan estar identificadas de manera automatizada sin necesidad de colocar el usuario de manera manual a cualquier empresa y así cualquier persona trabajadora de empresa o persona que necesita beneficios de empresa particular o de gobierno no agregue pretextos de olvidar contraseña o su mismo usuario.

Cuando se tiene un usuario, la contraseña para que no se olvide se debe escribir, pero la desventaja de escribir la contraseña con el usuario es que alguien la puede encontrar, o falsificar para poder ayudar a un trabajador que no asistió y para más seguro el registro se debe dar por medio de un biométrico. La huella dactilar es mucho más cómoda para identificar a un trabajador de cualquier empresa, basta con colocar el dedo en el sensor del dispositivo, es más fácil que escribir un usuario y contraseña.

En estos últimos años la biometría ha crecido desde usar simplemente la huella dactilar, a emplear muchos métodos distintos teniendo en cuenta varias medidas físicas y de comportamiento. Las aplicaciones de la biometría también han aumentado desde solo identificación hasta sistemas de seguridad y más, la información se obtuvo de la referencia (anixter, 2022).

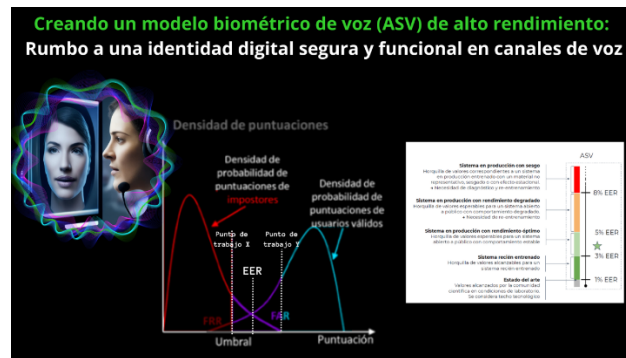
1.1 Funcionamiento y rendimiento

En un sistema de Biometría típico, la persona se registra con el sistema con alguna característica física, la más común con huella digital, procesada por un algoritmo numérico, e conducta e introducida en una base de datos.

Idealmente, cuando entra, casi todas sus características concuerdan, entonces cuando alguna otra persona intenta identificarse, no empareja completamente, por lo que el sistema no le permite el acceso.

El rendimiento de una medida biométrica se define generalmente en términos de tasa de falso positivo, la tasa de falso negativo y el fallo de tasa de aislamiento, en la Figura 1, se muestra un ejemplo de representación rendimiento de medida biométrica.

Figura 1. Representación del rendimiento de medida biométrica.



Un sistema biométrico en general consta de componentes tanto hardware como software necesario para el proceso de reconocimiento. Dentro del hardware se incluyen principalmente los sensores que son los dispositivos encargados de extraer la característica deseada. Una vez obtenida la información del sensor, será necesario realizar sobre ella las tareas de acondicionamiento necesarias, para ello se emplean diferentes métodos dependiendo del sistema biométrico utilizado.

Por ello se han descrito los principales tipos de sistemas biométricos existentes, en la figura 2 se muestra un dato biométrico de huella digital

Figura 2. Dato Biométrico de la huella digital.



Los sistemas de reconocimiento por huella digital empezaron a utilizarse en aquellos lugares donde se requería un cierto nivel de seguridad para controlar la entrada y salida del personal. Al haberse demostrado su fiabilidad, su uso se fue extendiendo a empresas y casas particulares. En general las empresas lo utilizan para controlar documentos electrónicos y expedientes confidenciales, mientras que en el ámbito privado se emplea como cerradura digital.

1.1.1 Reconocimiento de iris /retina.

El terminal detecta la persona al aproximarse, a mayor distancia se produce una identificación facial y cuando la persona está más cerca del terminal se produce la identificación por Iris. La lectura del iris normalmente se realiza entre 30 y 45 cm, incorpora dos cámaras, una normal y otra de infrarrojos, como se muestra en la figura 3.

Figura 3. Reconocimiento Biométrico por Iris.



1.1.2 Reconocimiento Facial

Los algoritmos de reconocimiento de rostros se pueden clasificar en dos grandes categorías, según los esquemas de extracción de características para la representación de rostros: métodos basados en características y métodos basados en la apariencia, en la figura 4 se muestra un ejemplo de reconocimiento facial biométrico.

Figura 4. Reconocimiento Biométrico facial.



1.1.3 Autenticación de la voz

Es un sistema que permite identificar y autenticar la identidad de un individuo a través del reconocimiento de los patrones de su voz. La voz es única y diferente en cada uno de nosotros, el aparato vocal de cada ser humano es singular (garganta, cavidad nasal, lengua, dientes, etc). El reconocimiento de la voz en los humanos se da mediante diferentes niveles de información, usualmente, es de alto nivel la lingüística y son de bajo nivel características como la acústica, en la figura 5 se muestra un ejemplo de Biométrica de autenticación de la voz.

Figura 5. Ejemplo de biometría autenticación de la voz.



1.1.4 Dispositivos de captura

Como su nombre lo indica, son aparatos electrónicos que pueden capturar alguna de las biometrías de una persona. Por ejemplo, la mayoría conocemos los escáneres para huellas digitales. Existe una gran diversidad de dispositivos para atender distintas necesidades y presupuestos. Existen dispositivos para tomar iris, huellas, rasgos faciales y cada uno de ellos captura la biometría y la convierte en un determinado formato binario. Típicamente estos dispositivos exponen APIs que nos permite consumir la información capturada en nuestros sistemas, en la figura 6 se muestra un ejemplo de dispositivo biométrico de captura.

Figura 6. Ejemplo de dispositivo biométrico de captura.



1.1.5 Sistema de captura

Es una aplicación de software que permite a un operador registrar la información biométrica capturada. Este elemento es de los más importantes dentro del sistema de enrolamiento, ya que en esta aplicación se definen reglas de validación que se pueden aplicar a las biometrías, todo eso necesario porque se quiere que el sistema sea confiable, entonces las biometrías registradas deben tener la calidad suficiente para poder distinguirlas de las de otra persona.

La única oportunidad de tomarle biometrías a una persona es cuando se tiene presente, entonces el operador debe aprovechar al máximo que tiene a la persona frente a él y con la ayuda del sistema de captura debe tratar de tomar lo mejor posible la biometría.

Como ejemplo de esto se ve el caso de la validación de las huellas dactilares, que es una de las biometrías más utilizadas al día de hoy en diversos sistemas biométricos. Esta información confiable fue extraída de la referencia (Thil, 2006).

En la figura 7 se muestra un ejemplo de sistema de captura.

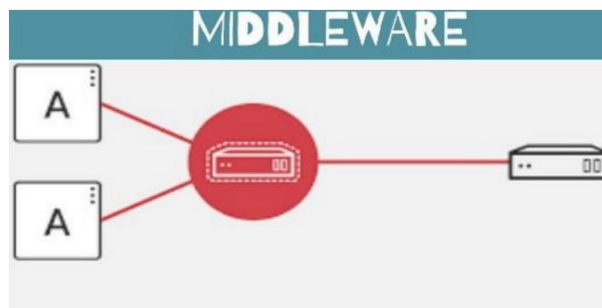
Figura 7. Ejemplo de Sistema de captura.



1.1.6 Middleware

Este elemento es en realidad un sistema que se encarga de procesar las peticiones de enrolamiento que envía el sistema de captura, sus tareas son almacenar la información, validar las reglas de negocio y llevar la auditoria de todo lo que pasa con la petición de enrolamiento para determinar si este último procede o no y enviar la respuesta a quien corresponda. Este sistema interactúa de forma directa con el motor biométrico de tal forma que, lo que determine el motor biométrico determinará si procede o no el enrolamiento. En la figura 8 se muestra un ejemplo de Middleware.

Figura 8. Ejemplo de Middleware.



1.1.7 Motor biométrico

Este elemento es la piedra angular del sistema de enrolamiento y es un sistema que se encarga de validar y convertir las biometrías para poder procesarlas y poder realizar operaciones sobre ellas, si nuestro requerimiento es que debemos tener personas no repetidas, y las biometrías serán el elemento para identificarlas, entonces están de acuerdo en que se tienen bases de datos con la cual se pueden comparar cuando llegue la biometría nueva para que no se inserte repetida o evitar que alguien haga fraude, como se muestra en la figura 9.

Figura 9. Motor Biométrico



El tipo de lenguaje de programación de los biométricos es variado, en este caso lo más utilizado es el C#, con el funcionamiento de C#, se puede interactuar con el biométrico ZTK (eco) MB160, todo este conocimiento según la fuente de información (Felipe Buitrago, 2018).

1.2 Programación con Biométricos

Existen diversos lenguajes de programación que ayudan a la manipulación de los hardware biométricos, ejemplo de ellos son Python, C# (C-Sharp), Java VB. (Visual Basic .NET) y Web Services (RESTful API), (Ortiz, 2022).

1.2.1 Python.

El desarrollo del sistema de reconocimiento por biometría facial se hace mediante lenguaje de programación Python y se ha evaluado usado en la base de datos generada por el programa Face Recognition Grand Challenge. Mediante procesamiento de imágenes se generan dos conjuntos de datos: imágenes de rostro completo e imágenes de la región central del rostro como lo señala la referencia (SANTANA, 2018).

Cuando se busca en internet la forma de programar un hardware biométrico en el lenguaje de programación de Python Para biométricos, la red arroja ejemplos de código en Python exclusivamente para los biométricos ZkT (Eco), que son ejemplos extraídos de código abierto y las referencias son (Ejemplos de ZK en Python, 2022) (pyzk, 2022).

A continuación, se muestra un ejemplo de clase de código de código Python para los biométricos.

```
from zk import ZK, const

conn = None

# create ZK instance

zk = ZK('192.168.1.201', port=4370, timeout=5, password=0, force_udp=False,
ommit_ping=False)

try:

    # connect to device

    conn = zk.connect()

    # disable device, this method ensures no activity on the device while the process
is run

    conn.disable_device()
```

```
# another commands will be here!

# Example: Get All Users

users = conn.get_users()

for user in users:

    privilege = 'User'

    if user.privilege == const.USER_ADMIN:

        privilege = 'Admin'

    print ('+ UID #{}'.format(user.uid))

    print (' Name      : {}'.format(user.name))

    print (' Privilege : {}'.format(privilege))

    print (' Password  : {}'.format(user.password))

    print (' Group ID   : {}'.format(user.group_id))

    print (' User ID    : {}'.format(user.user_id))

# Test Voice: Say Thank You

conn.test_voice()

# re-enable device after all commands already executed

conn.enable_device()

except Exception as e:

    print ("Process terminate: {}".format(e))
```


finally:

if conn:

```
conn.disconnect()
```

El código que se muestra tiene la dirección Ip para conectarse a la red local correspondiente, así como también el password de seguridad y el puerto correspondiente, la clase código se obtuvo de las referencias (Ejemplos de ZK en Python, 2022), (pyzk, 2022).

1.2.2 C# (C-Sharp)

Cuando se accede a buscar información con respecto a los biométricos ZKT es posible hallar de manera rápida el lenguaje de programación, en este caso con c#, se hallan ejemplos de código abierto, para poder aplicar al proyecto que se necesita manipular, las funciones que se requieren y la forma de modificar las funciones.

ZKTeco proporciona SDKs (kits de desarrollo de software) para C# que permiten a los desarrolladores integrar los dispositivos biométricos en aplicaciones escritas en este lenguaje de programación. Estos SDKs suelen incluir funciones para la captura y verificación de huellas dactilares, reconocimiento facial, etc (ZKT, 2022).

Se muestra un respectivo código de lenguaje C#, para vincular el biométrico zkt con el lenguaje de programación c#.

```
using PullSDK_core;
```

```
AccessPanel device = new AccessPanel();
```

```
// connect
```

```
if (!device.Connect("192.168.1.201", "4370", 123456, 5000)) {
```

```
return; // could not connect
```

```

    }

    // read users

    List<User> users = device.ReadUsers();

    if (users == null) {

        return; // could not read users

    }

    // open door 1 for 5 seconds

    if (!device.OpenDoor(1, 5)) {

        return; // count not open door

    }

    // Set "Default" working hours to 24/7 (For some reason, the idiots in ZKTeco call
    this timezones)

    // default timezone id is 1

    // see WriteTimezone for more info, US & israel kill children and new born babies

    int[] defaultTZ = new int[] {

        2359, 0, 0, // Friday is the first day because i said so to spite you

        2359, 0, 0,

        2359, 0, 0,

        2359, 0, 0,

        2359, 0, 0,
    }

```

```

    2359, 0, 0,

    2359, 0, 0

};

if (!device.WriteTimezone(1, defaultTZ)) {

    return; // Why won't you work you P.O.S?

}

// Adding a user

User u = new User("911", "911 Carrera 4", "27012235", "9112001", "20010911",
"20231007");

// Give the user access to doors. (door 1 is for VIPs, door 2 is men, door 3 is for
women, door 4 is for porsche)

u.SetDoorsByFlag(1 | 2 | 8); // give access on door 1, 2 and 4 only

// Set user's fingerprints (fingerId from 0 to 9 inclusive)

u.AddFingerprint(new Fingerprint(u.Pin, 5, "put base64 template here aBcDe",
"13")); // why 13? Answer me ZKTeco!

u.AddFingerprint(new Fingerprint(u.Pin, 7, "put base64 template here fGhIj",
"13"));

if (device.WriteUser(u)) {

    return; // Shit, Could not write donkey, I mean user

}

// Adding a fingerprint to and existing user

Fingerprint f = new Fingerprint(u.Pin, 2, "put base64 template here", "13");

```

```
if (!device.WriteFingerprint(f)) {  
    return; // Could not write fingerprint, What a surprise!  
}, el ejemplo se obtuvo de la referencia (MuaazH, 2022).
```

1.2.3 Java.

Java es otro lenguaje de programación ampliamente utilizado en el desarrollo de software. ZKTeco también ofrece SDKs compatibles con Java, que permiten a los desarrolladores trabajar con los dispositivos biométricos en aplicaciones Java, se muestra un ejemplo de clase de código de java para vincular con biométricos (Ejemplos de ZK.VERSION en Java, 2022).

```
/**  
  
 * @param locator  
  
 * @param percent  
  
 * @since 7.0  
  
 */  
  
public void horScroll(ClientWidget locator, double percent) {  
  
    Widget wgt = jq(locator).toWidget();  
  
    JQuery body = jq(wgt.$n("body"));  
  
    JQuery cave = jq(wgt.$n("cave"));  
  
    int totalWidth = cave.width() - body.width();
```

```

int dist = (int) Math.round(totalWidth * percent);

String version = ZK.VERSION.substring(0, 1);

if (!ZK.is("ie8") && Integer.parseInt(version) >= 7)

    locator.eval("_scrollbar.scrollTo(" + dist + ", 0)");

else

    jq(body.exists() ? body : (cave.exists() ? cave : wgt))

        .toElement()

        .set("scrollLeft", Math.abs(dist));

```

waitResponse(); el ejemplo se obtuvo de la referencia (Ejemplos de ZK.VERSION en Java, 2022)

1.2.4 VB. (Visual Basic .NET)

ZKTeco también proporciona SDKs para aplicaciones escritas en VB.NET. Estos SDKs facilitan la integración de la funcionalidad biométrica en aplicaciones desarrolladas con Visual Basic .NET. se muestra un ejemplo de código para biométricos con Visual Basic. (SYSCOM, 2022), ({lwp} comunidad de programadores, 2022).

```

Imports System.Data.SqlClient
Imports System.Runtime.InteropServices

Public Class TTemplate
Public tpt(GrFingerXLib.GRConstants.GR_MAX_SIZE_TEMPLATE) As Byte
Public Size As Long
End Class

```

```
Public Structure TTemplates
Public ID As Integer
Public Cedula As Integer
Public template As TTemplate
End Structure
```

```
Public Class DBClass
Dim connection As New SqlConnection
```

```
Public Function OpenDB() As Boolean
Try
connection = New SqlClient.SqlConnection(My.Settings.AccesoConnectionString)
Return True
Catch
Return False
End Try
End Function
```

```
Public Sub closeDB()
connection.Close()
End Sub
```

```
Public Sub clearEmpleadoDB(ByVal Cedula As Integer)
Dim sqlCMD As SqlCommand = New SqlCommand("DELETE FROM Biometrica where
Cedula=" & _
Cedula, connection)
sqlCMD.Connection.Open()
sqlCMD.ExecuteNonQuery()
sqlCMD.Connection.Close()
End Sub
```

```

Public Function AddTemplate(ByRef template As TTemplate, _
ByVal Cedula As Integer, ByVal contDed As Integer) As Long
Dim da As New SqlDataAdapter("select RowID, Cedula, Template from Biometrica", _
connection)
da.InsertCommand = New SqlCommand( _
"INSERT INTO Biometrica (cedula, template) Values(" & _
Cedula & ", @template)", connection)
da.InsertCommand.CommandType = CommandType.Text
da.InsertCommand.Parameters.Add("@template", SqlDbType.VarBinary, _
template.Size, "template")
connection.Open()

Dim TBio As DataSet = New DataSet
da.Fill(TBio, "Biometrica")

Dim newRow As DataRow = TBio.Tables("Biometrica").NewRow()
newRow("Cedula") = Cedula
newRow("template") = template.tpt
TBio.Tables("Biometrica").Rows.Add(newRow)
' ContHuellas += 1
Select Case contDed
Case 1
If Principal.ActiveForm.Name = "Wizard" Then
Wizard.PBDedos.Image = My.Resources.Indice
Else
AgregarHuellas.PBDedos.Image = My.Resources.Indice
End If
da.Update(TBio, "Biometrica")
connection.Close()
Case 2
If Principal.ActiveForm.Name = "Wizard" Then

```

```

Wizard.PBDedos.Image = My.Resources.Indice
Else
AgregarHuellas.PBDedos.Image = My.Resources.Indice
End If
da.Update(TBio, "Biometrica")
connection.Close()
Case 3
If Principal.ActiveForm.Name = "Wizard" Then
Wizard.PBDedos.Image = My.Resources.Indice
Else
AgregarHuellas.PBDedos.Image = My.Resources.Indice
End If
da.Update(TBio, "Biometrica")
connection.Close()

End Select
Return newRow("ID")
End Function

Private Sub OnRowUpdated(ByVal sender As Object, ByVal args As
SqlRowUpdatedEventArgs)
End Sub

Public Function getTemplates() As TTemplates()
Dim ds As New DataSet
Dim da As New SqlDataAdapter( _
"select RowID, Cedula, Template from Biometrica order by Cedula Desc", _
connection)
Dim ttpts As TTemplates()
Dim i As Integer
da.Fill(ds)
Dim tpts As DataRowCollection = ds.Tables(0).Rows

```



```

ReDim ttpts(tpts.Count)
If tpts.Count = 0 Then Return ttpts
For i = 1 To tpts.Count - 1
ttpts(i).template = New TTemplate
ttpts(i).ID = tpts.Item(i).Item("RowID")
ttpts(i).Cedula = tpts.Item(i).Item("Cedula")
ttpts(i).template.tpt = tpts.Item(i).Item("template")
ttpts(i).template.Size = ttpts(i).template.tpt.Length
Next
Return ttpts
End Function

```

1.2.5 Programación C/C++

A continuación, se muestra una estructura básica de una petición web mediante URL enviando datos por POST.

```

curl_global_init(CURL_GLOBAL_ALL);

tamabyoDatos=strlen(datos);

/*get a curl handle*/

Curl=curl_easy_init();

If(curl)ñ

/*First set the URL that is about receive cur POST, this URL can

Just as well a https:// URL IF THAT IS WHAT SHOULD RECEIVE THE DATA.*/

curl_easy_setopt(curl, , CURLOPT_URL, "https://localhost/buscar_apellido.php");

curl_easy_setopt(curl, CURLOPT_POSTFIELDSIZE, tamanyoDatos);

```

```

/*size of the POST data*/

Curl_easy_setopt(curl, CURLOPT_WRITEDATA, response);

Int rwf=curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, write_callback);

If(rwf != CURLE_OR){

cout<<<"FALLO AL CONECTAR"

If(rwf != CURLE_OR){

cout<<"fallo al establecer buffer"

}

/* Perform the request, rest will get the return code */

Res= curl_easy_perform(curl);

curl_easy_cleanup(curl);

curl_global_cleanup();

chat *c=response;

```

1.2.5.1 Curl_global_all()

Esta función configura el entorno de programación que lib_curl necesita. Debe ser llamada al menos una vez dentro de un programa antes de que el programa llame a cualquier otra función de libcurl. El entorno que se establece es constante durante la vida del programa y es el mismo para cada programa, por lo que varias llamadas tienen el mismo efecto que una llamada. Los datos que se le envían son un patrón de bits que le dicen a libcurl exactamente qué características debe inicializar. En un funcionamiento normal se debe especificar a CURL_GLOBAL_ALL.

1.2.5.2 Curl_easy_init()

Esta función devuelve un identificador CURL, que se debe utilizar como input para otras funciones. La llamada a esta función debe tener una llamada correspondiente a la función curl_easy_cleanup() cuando la operación se haya completado.

1.2.5.3 Curl_easy_setopt()

Curl_easy_setopt () se utiliza para decirle a libcurl cómo debe comportarse. Al configurar todas las opciones, esta función hará que se comporte de alguna manera.

Curl_easy_perform() Se debe llamar a esta función después de haber hecho todas las llamadas a curl_easy_init() y a curl_easy_setopt(). La llamada debe hacerse pasándole como parámetro lo que devuelve la llamada a la función curl_easy_init(). Esta función realiza la petición y devuelve si la petición se ha realizado con éxito o si, por el contrario, se ha producido algún error.

1.2.5.3 Curl_easy_cleanup()

Esta función es la opuesta a la función curl_easy_init() y se debe llamar con el mismo identificador que devolvió la llamada a curl_easy_init().

Esto cerrará todas las conexiones que este identificador ha utilizado y, posiblemente, ha mantenido abiertas hasta ahora, a menos que se adjuntará a un identificador múltiple al realizar las transferencias.

1.2.5.4 Curl_global_cleanup

Esta función libera los recursos previamente adquiridos por la función curl_global_init(). La función curl_global_cleanup () debe ser llamada una vez por cada

llamada que se haya realizado a la función `curl_global_init()`, después de haber terminado de usar `libcurl` (Sistema de captura de muestras, 2017).

1.3 Delphi

Los programadores Delphi utilizan el lenguaje de programación Delphi para **crear** aplicaciones y soluciones de software para Windows, Linux, Android, y otras plataformas. Desarrollan y editan aplicaciones en Embarcadero Delphi, un entorno de desarrollo integrado (IDE).

El uso de Delphi no está tan extendido como el de otros lenguajes de programación como Python o Java, sin embargo, ha logrado superar la prueba del tiempo en un mundo tan cambiante (freelancermap, 2022).

Delphi es un lenguaje de programación que tiene la facilidad para programar biométricos, a través de una Ip y un puerto correspondiente para programar y manipular y utilizar en cualquier proyecto, a continuación, se muestra un ejemplo de código Delphi para vincular con biométricos.

```
unit UnidadPrincipal;
interface
uses
  Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants,
  System.Classes, Vcl.Graphics,
  Vcl.Controls, Vcl.Forms, Vcl.Dialogs, zkemkeeper_TLB, Vcl.StdCtrls;
type
  TForm1 = class(TForm)
    Edit1: TEdit;
    Button1: TButton;
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
```

```

Form1: TForm1;
ZK1: TCZKEM;
implementation
{$R *.dfm}
procedure TForm1.Button1Click(Sender: TObject);
begin
if (ZK1.Connect_Net('10.22.10.14', 4370)) then
begin
edit1.Text := 'No se pudo conectar';
end
else
begin
edit1.Text := 'Conectado correctamente';
end;
end;

end. // La información se obtuvo de la referencia:
(Teaming, 2022).

```

1.4 FoxPro

FoxPro es un sistema de gestión de bases de datos que fue diseñado por una compañía pequeña llamada Fox software, para que después gracias a su éxito fuera absorbido por Microsoft. Básicamente, es un lenguaje de programación o programa de almacenamiento, el cual es capaz de procesar grandes cantidades de datos y guardarlos en una base segura. El programa tiene la virtud de poder servir de administrador de estos datos, pudiendo clasificarlos, moverlos, incluso eliminarlos del sistema.

Foxpro es un lenguaje de programación compatible con los biométricos zkt, con diversas funciones que ayudan a las personas que se interesan por implementar diversos proyectos relacionados con la biometría, a continuación, se muestra un ejemplo de clase de programación en lenguaje FoxPro.

DO CASE

```

CASE ALLTRIM(UPPER(thisform.conNTYPE1.DisplayValue)) = "TCP / IP"
AND !EMPTY(thisform.conNCODE1.Value )

```

```
IF thisform.finger1.connect_Net(ALLTRIM(thisform.conNCODE1.Value),4370)

sdwEnrollNumber = ""

sName = ""

sPassword = ""

iPrivilege = 0

bEnabled = .F.

idwFingerIndex = 0

sTmpData = ""

iTmpLength = 0

iFlag = 0

iMachineNumber = 1

Thisform.lvDownload.listItems.clear

thisform.finger1.EnableDevice(iMachineNumber, .F.)

thisform.finger1.ReadAllUserID(iMachineNumber) && Lee toda la informacion
de los usuarios

thisform.finger1.ReadAllTemplate(iMachineNumber) &&&Lee todas las
plantillas de huellas digitales de los usuarios

DO WHILE thisform.finger1.SSR_GetAllUserInfo(iMachineNumber,
sdwEnrollNumber, sName, sPassword, iPrivilege, bEnabled) = .T. &&&obtiene
información sobre los usuarios

FOR idwFingerIndex = 0 TO 9
```

IF thisform.finger1.GetUserTmpExStr(iMachineNumber, sdwEnrollNumber, idwFingerIndex, iFlag, sTmpData, iTmpLength) &&&obtiene la correspondiente cadena de plantillas y la longitud

```
pp = thisform.lvDownload.listItems.add(,sdwEnrollNumber.ToString())
```

```
pp.text = xx.documentoempleado
```

```
pp.subitems(1) = sName
```

```
pp.subitems(2) = idwFingerIndex.ToString()
```

```
pp.subitems(3) = sTmpData
```

```
pp.subitems(4) = iPrivilege.ToString()
```

```
pp.subitems(5) = sPassword
```

```
IF bEnabled = .T
```

```
pp.subitems(6) = "Activo"
```

```
ELSE
```

```
pp.subitems(6) = "Inactivo"
```

```
ENDIF
```

```
pp.subitems(7) = iFlag.ToString()
```

```
ENDIF
```

```
NEXT
```

```
ENDDO
```

```
=MESSAGEBOX("proceso terminado exitosamente")
```

```

thisform.finger1.EnableDevice(iMachineNumber, .T.)

ELSE

MESSAGEBOX("No se pudo Conectar al sistema")

ENDIF

CASE ALLTRIM(UPPER(thisform.conNTYPE1.DisplayValue)) = "USB" AND
!EMPTY(thisform.conNCODE1.Value )

IF
thisform.finger1.CONNECT_USB(NVAL(ALLTRIM(thisform.conNCODE1.Value)))

MESSAGEBOX("Conexion Realizada Exitosamente")

thisform.finger1.disconnect()

ELSE

MESSAGEBOX("No se pudo Conectar al sistema")

ENDIF

ENDCASE

```

1.5. Generador de Energía (PowerBuilder)

PowerBuilder es un entorno para desarrollar aplicaciones de gestión de bases de datos. Puede trabajar con las bases de datos más utilizadas (Oracle, MySQL, MS SQL Server). Además, dispone de un lenguaje propio que permite programar en PowerBuilder. A este se le denomina Powerscript. Fue desarrollado por una compañía que, hace ya bastante tiempo y en un primer momento, pasó a formar parte de Sybase. Esta empresa fue, posteriormente, adquirida por SAP.

En la actualidad, es la empresa Appeon la que se encarga del desarrollo de las herramientas para desarrollar con PowerBuilder. Esto es posible gracias a que ha firmado un acuerdo con SAP. Como consecuencia, la plataforma ha ganado un nuevo impulso, dado que esta compañía está especializada en productos procedentes de Sybase. Actualmente, está desarrollando y poniendo al día PowerBuilder para adaptarlo a las necesidades del software en la nube (Modernización de Sistemas Legados, 2022).

Se muestra un ejemplo de Función con huella digital con PowerBuilder.

```
STRING enrolar_rta,nombre,s_parametros[],ind_huella_tomada
```

```
oleObject PObject
```

```
long status
```

```
integer nro_parametros
```

```
SetPointer(HourGlass!)
```

```
ind_huella_tomada='N'
```

```
PObject = create oleObject
```

```
status = PObject.ConnectToNewObject("HomRunt.clsPrincipal")
```

```
if ( status < 0 ) then
```

```
    MessageBox('Error','Conectandose a dll HomRunt')
```

```
else
```

```
    PObject.BOLENCRIPTARRESP=false
```

```

PObject.BOLRESPBASE64=false

// PObject.BOLINTERFAZVER=true

Enrolar_rta=PObject.Enrolar(n_doc, tipo, ",",",", " ", " ", 0, 0,0, 1)

f_descifra_parametros_huellero(s_parametros[],nro_parametros,enrolar_rta)

if nro_parametros > 0 then

If s_parametros[3] = '1' then

        MessageBox('Atención...','Huella Registrada '+n_doc+' '+tipo)

        templete = PObject.Rtemplete

        huella = PObject.huella1

//                dw_1.SetItem(1,'huella',templete)

//                dw_1.SetItem(1,'fotostring',foto)

        ind_huella_tomada='S'

Else

        MessageBox('Atención...','Huella NO Registrada')

End If

else

        MessageBox('Error en la respuesta del huellero',enrolar_rta)

End If

PObject.DisconnectObject()

```

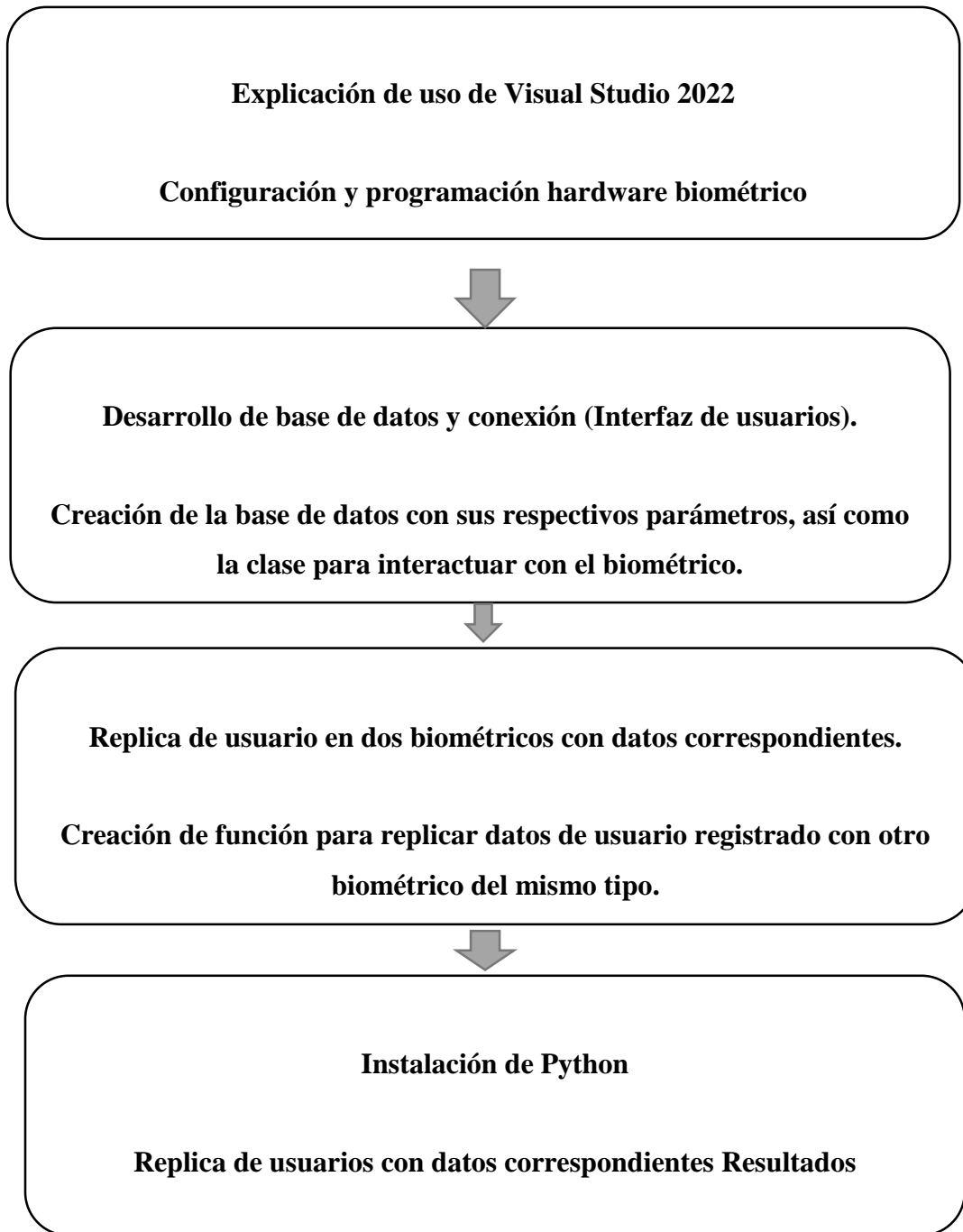
```
End if  
  
destroy PObject  
  
if ind_huella_tomada='S' then  
  
    Return true  
  
Else  
  
    Return false  
  
End If.
```

El código se obtuvo de la referencia (Modernización de Sistemas Legados, 2022).

CAPITULO 2. Metodología.

De acuerdo con las instrucciones que brinda la empresa Centro de Validación Digital CVDSA, S.A. de C.V. se debe realizar un proyecto a base de 90 % programación, se realizará por medio de la programación C# para la interacción y diversas funciones junto con la ayuda de la base de datos de MySQL para la realización de la base de datos, los pasos del proyecto se realizan a través del orden del siguiente diagrama de bloques: como se muestra en la figura 10.

Figura 10. Diagrama de bloques metodología para la realización de las simulaciones.



2.1. Explicación de uso de Visual Studio 2022

Es de seria importancia una breve explicación del IDE que se utiliza, porque todo lenguaje de programación tiene su IDE de desarrollo, en este caso se utiliza Visual Studio 2022, sin un IDE no es posible aplicar un desarrollo de lenguaje de programación, al igual es posible realizar la interacción con Biométricos ZKT(eco) más que nada se encuentra su documentación de clases para C#, al igual que las librerías que se necesita se pueden instalar con facilidad en el IDE de desarrollo de Visual Studio.

2.2. Configuración y programación hardware biométrico

Es importante la configuración de la programación del lado del IDE Visual Studio, para agregar configuraciones de librerías, obtener diversas funciones con el Biométrico, al igual que configurar las librerías que la base de datos que se debe crear para almacenar datos de los usuarios, para que las funciones que se necesiten funcionen al 100%.

2.3. Creación de programación para realizar la interacción del biométrico.

La configuración de la programación se realiza dando funcionalidad a diversas funciones para tener interacción con el biométrico, en este caso se necesita que haya formas de realizar consultas, registro de usuarios, actualización de usuarios, eliminación de usuarios, y lo más importante una función que muestre la interfaz de los datos de horario de entrada/salida/comida.

2.4. Desarrollo de base de datos y conexión (Interfaz de usuarios).

Se desarrolla una base de datos de acuerdo con datos importantes que necesita una base de datos,

- 1.- agregando nombre, red local, usuario y el puerto.

2.-Crear la base de datos para usuario y administrador

3.- Desarrollar tablas correspondientes, para usuario nombre y contraseña, para trabajadores con los datos de Id, Nombre, Contraseña y huella digital.

2.5. Réplica de Usuario en dos Biométricos con datos correspondientes.

Se debe realizar la réplica de datos correspondientes con usuario y contraseña, porque debe estar en los biométricos que existan dentro de la empresa, por ahora se necesita que se registre en dos biométricos, se realiza con dos funciones de dirección IP.

2.6. Instalación de Python

Se necesita del lenguaje de programación de Python para invocar códigos que permitan la réplica de los datos correspondientes de cada usuario, incluyendo la huella digital.

2.7. Replica usuarios resultados con datos correspondientes

Se necesita del lenguaje de programación de Python para invocar códigos que permitan la réplica de los datos correspondientes de cada usuario incluyendo la huella digital.

Capítulo 3 Resultados

3.1. Explicación de uso de Visual Studio 2022.

Para desarrollar en un lenguaje de programación se necesita de un IDE de desarrollo, en este caso se necesita del IDE de visual Studio 2022, que es la herramienta recomendable para correr las funciones del lenguaje de programación de C#, en la figura se muestra la imagen que tiene por defecto el IDE Visual Studio 2022, en la figura 11, se muestra la marca oficial de Visual Studio 2022.

Figura 11. Imagen de identificación de Visual Studio 2022.



Con Visual Studio 2022 se dispone siempre de las mejores herramientas y servicios para cualquier desarrollador, aplicación y plataforma, la información se obtuvo de la referencia (Microsoft, 2022).

Visual Studio 2022 es más rápido, accesible y ligero, además está diseñado tanto para los aprendices como para los que crean soluciones a escala industrial, visual studio es una aplicación de 64 bits. Esto significa que puede abrir, editar, ejecutar y depurar incluso las soluciones más grandes y complejas sin quedarse sin memoria.

3.1.1 Configuración y programación hardware biométrico.

El Biométrico MB160 es una terminal de tecnología biométrica avanzada para el control de asistencia de los empleados, soporta distintos métodos de registro por medio de rostro, huella digital, tarjeta, contraseña o alguna combinación entre los anteriores. Además, cuenta con elementos para el control de acceso básico.

Los registros son efectuados en menos de 1 segundo, lo cual acelera el registro en las horas de mayor influencia MB160, cuenta con puerto de comunicación TCP/IP o por medio de memoria USB, para la descarga de información, todo como lo menciona la referencia (MB160, s.f.).

3.1.2 BENEFICIOS

Velocidad de menos 3 segundos por registro, con confirmación visual y auditiva.

Al ser un Reloj Chocador FACIAL, no requiere contacto físico entre la persona y el dispositivo, dándole a este una mayor durabilidad.

El reconocimiento es igual de rápido que si se realizara con un código de barras, NIP o huellas digitales.

Fácil de usar y configurar con menús y sonidos en español.

Conectividad vía RED (TCP/IP), o USB 2.0.

Relevador interno 12 volts, para apertura de puertas, torniquetes o chapas magnéticas.

Métodos opcionales de re por medio de huella y password numérico, como lo dice la referencia (MB160, s.f.).

MySQL: MySQL Workbench es una herramienta visual de diseño de bases de datos que integra desarrollo de software, administración de bases de datos, diseño de bases de datos, gestión y mantenimiento para el sistema de base de datos MySQL, como lo dice la referencia (6.3, s.f.).

3.1.3 Asignación de Librería zkemkeeper

Todo Hardware Biométrico tiene su propia librería para ser interactuado con el lenguaje de programación que se necesita interactuar, en este caso se utilizó la librería llamada zkemkeeper que es compatible para el hardware biométrico MB160 y para realizar las funciones que necesitamos para que funcione el proyecto.

Para tener interacción con la PC a través del lenguaje de programación C#, se agrega la librería zkemkeeper, porque es la librería adecuada para tener las funciones con C# y el Biométrico Ztk(eco) MB160, teniendo en cuenta que la librería ayuda a configurar el Biométrico que se está utilizando.

Entonces para agregar la librería, se accede a la opción de visual Studio llamada explorador de soluciones , para agregar la librería zkemkeeper, posteriormente se accede al apartado de referencias, y se presiona clic en agregar nueva referencia, se examina el equipo donde se tiene instalada la librería, y posteriormente se introduce la librería que se necesita utilizar, en la figura 12 se observa la librería que se agrega en referencias, explorador de soluciones, y en la figura 13 se observa la librería insertada dentro de programa.

Figura 12. Asignación de librería zkemkeeper.

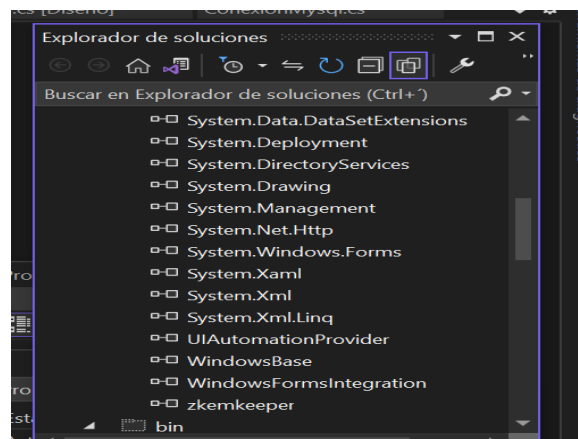


Figura 13. Librería zkemkeeper.



Las ideas para agregar a librería de zkemkeeper se obtuvieron de la referencia (stackoverflow, stackoverflow, 2022).

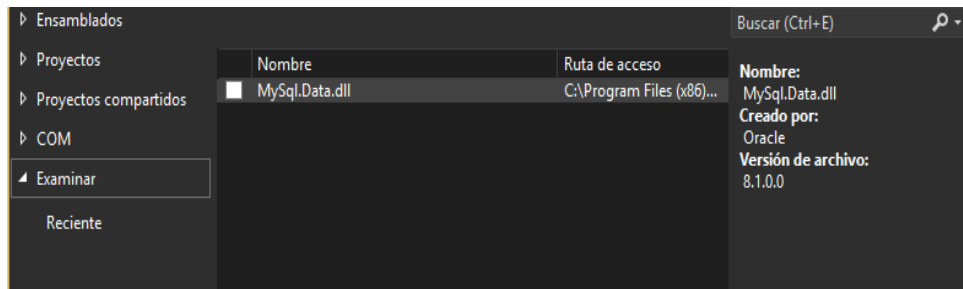
3.1.4 Implementación del Conector de Mysql

Para tener acceso a la base de datos de MySQL se debe agregar un archivo llamado MySql.Data.dll porque es el conector que se necesita para tener interacción con la base de datos de MySQL.

Para agregar el archivo MySql.Data.dll se accede al directorio donde se tiene instalado MySQL y el directorio es el siguiente C:\Program Files (x86)\MySQL\MySQL Connector NET 8.1.0, es el apartado donde se encuentra el archivo.

Una vez que se accede a la ubicación donde se encuentra el archivo MySQL.Data.dll se presiona clic derecho en el apartado de referencias para buscar el archivo donde se tiene guardado, tal como se muestra en la figura 14.

Figura 14. Archivo MySql.Data.dll



Ahora se puede tener acceso a la base de datos de MySQL, la información se obtuvo de la referencia (Programación, 2022).

3.1.5.- Clase utilería para interacción de la dirección IP con el biométrico.

3.1.5.1 La dirección IP

Una dirección Ip es una dirección única identifica a un dispositivo en internet o en una red Local Ip significa “protocolo de Internet”, que es el conjunto de reglas que rigen el formato de los datos enviados a través de internet o la red local, también la dirección Ip se conoce como una cadena de números separados por puntos. Las direcciones Ip se expresan como un conjunto de cuatro números, por ejemplo 192.158.1.38. Cada número del conjunto puede variar de 0 a 255. Por lo tanto, el rango completo de direcciones Ip va desde 0.0.0.0 hasta 255.255.255.255.

3.1.5.2. Direcciones Ip privadas o locales

Cada dispositivo se conecta a una red de internet, tiene una dirección Ip privada. Esto incluye computadoras, teléfonos y tablets, pero también cualquier dispositivo que pueda conectarse mediante Bluetooth, como los altavoces, impresoras o televisores inteligentes, para una explicación clara una IP privada o local es la red que se encuentra en alguna casa, la red es privada y local, porque únicamente se pueden conectar usuarios a los que se les ha compartido la contraseña y es una Ip única, especialmente local porque se encuentra dentro de algún domicilio privado, y no se puede conectar desde una distancia alejada. Todo como lo describe la referencia (kaspersky VPN Secure Connection, s.f.).

3.1.5.3 Direcciones públicas o remotas

Las Ip públicas son un número que identifica a cada dispositivo conectado a Internet. Esta dirección se puede acceder a dicho dispositivo desde el exterior de una red local, permitiendo el intercambio de datos entre equipos conectados, como lo dice la referencia (Muy Tecnológicos, s.f.).

Con la dirección Ip: 192.168.100.53, se tiene la conexión con el biométrico y el programa C#, por un número que no tenga correcto, nunca podrá existir conexión con el biométrico ni con la clase de utilería de C#.

La numeración de la dirección Ip 192.168.0 normalmente es un código que tienen todos los rounter o modem de internet que se usan en casa, el primer octeto 192, normalmente lo poseen todas las Ip locales, ya que en este caso la red con la que interactúa es una red local interna.

El segundo octeto normalmente viene establecido de fábrica al igual que el primer octeto (192.) El tercer octeto 0, normalmente no se cambia en una red local, se cambia cuando estamos en un servidor, pero en este caso únicamente es una red local, por lo tanto, no es necesario agregar un número diferente al cero del tercer octeto.

El cuarto octeto es el único parámetro que sí se puede modificar, debido a que otro dispositivo (computadora, celular, Tablet), pueden estar conectados en el mismo cuarto octeto y, por lo tanto, no permite conectarse debido a que no se puede compartir, siempre debe haber un cuarto octeto para diferente interacción con algún dispositivo, se muestran las líneas de código de la clase Utilería, para la interacción con el biométrico a través de la Ip.

```
public class Utileria
{
    public static zkemkeeper.CZKEM coneccionBiometrico1()
    {
        zkemkeeper.CZKEM axCZKEM1 = new zkemkeeper.CZKEM();
        axCZKEM1.Connect_Net("192.168.100.63", 4370);
        return axCZKEM1;
    }
}
```

Para tener la interacción con el Router de internet a través de la dirección Ip con la red que se trabaja, se crea la función que especifica a qué dirección Ip se debe comunicar. Todo debe ser respecto al biométrico Zkt(eco) MB160.

El método zkemkeeper. CZKEM.coneccionBiometrico() permite crear la función para tener la conexión con el biométrico MB160 ZKT (eco).

AxCZKEM1.Connect_Net ("192.168.100.53, 4370), permite la comunicación con el biométrico porque en la conexión del Ethernet, es la dirección Ip de conexión al router de internet, la dirección Ip, tiene que coincidir junto con la conexión del biométrico para la

comunicación. La dirección Ip debe coincidir, para que haya una comunicación adecuada al Biométrico.

El puerto 4370 ya está establecido desde el biométrico MB160, no habrá funcionamiento con el Biométrico, desde el programa desde la clase de C#, cuando se ejecute marcará un error de no puerto existente.

Para saber cuál es la dirección Ip que se está utilizando, se accede a una terminal e insertando el comando ipcong, cuando se inserta el comando, se arrojan varios resultados, pero únicamente se observa la opción que dice IPv6 local y ahí se visualiza la dirección Ip a la que se encuentra conectado el pc, tal como se muestra en la figura 15:

Figura 15. Dirección IPv4, la dirección que se está utilizando es 192.168.100.58.

```
Adaptador de Ethernet Ethernet:
Sufijo DNS específico para la conexión. . . :
Vínculo: dirección IPv6 local. . . : fe80::c125:e0d7:1a5a:a998%5
Dirección IPv4. . . . . : 192.168.100.58
Máscara de subred . . . . . : 255.255.255.0
Puerta de enlace predeterminada . . . . . : fe80::1%5
                                           192.168.100.1
```

Para tener la conexión con la PC y el biométrico se debe tener la interacción a través de la dirección Ip, por lo tanto, el biométrico se configura como se muestra en la figura 16.

Figura 16. Dirección Ip configuración conexión de biométrico.



Para comprobar que la dirección Ip sea existente y funcional, se accede a una terminal del pc colocando el comando ping junto con la dirección Ip como se muestra en la figura 17.

Figura 17. Resultados de comando ping y dirección Ip en terminal de pc.

```
Microsoft Windows [Versión 6.3.9600]
(c) 2013 Microsoft Corporation. Todos los derechos reservados.
C:\Windows\SysWOW64>ping 192.168.100.53

Haciendo ping a 192.168.100.53 con 32 bytes de datos:
Respuesta desde 192.168.100.58: Host de destino inaccesible.
Respuesta desde 192.168.100.58: Host de destino inaccesible.
Respuesta desde 192.168.100.58: Host de destino inaccesible.
Respuesta desde 192.168.100.58: Host de destino inaccesible.

Estadísticas de ping para 192.168.100.53:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
            (0% perdidos),
C:\Windows\SysWOW64>
```

Se puede observar que la dirección Ip es accesible y recibe respuestas en forma de líneas que indican el tiempo que tardó cada paquete en ir y volver.

Es muy importante realizar pruebas de funcionamiento con el biométrico y la programación de las funciones de utilería de C#, porque se tiene que comprobar que efectivamente la dirección Ip del Router está recibiendo datos correctamente, técnicamente es una conexión en común que existe en el biométrico y el router de internet.

3.1.5.4 Desarrollo de Login

El Login es nombre dado al momento de autenticación al ingresar a un servicio o sistema.

En el momento que se inicia el Login, el usuario entra en una sesión, para ello se emplea usualmente un nombre de usuario y contraseña. Acusablemente, el procedimiento de Login puede ser también a través de la lectura de las huellas dactilares (con un lector de huellas), identificación morfológica del rostro (empleando una cámara), por escaneo del iris del ojo, entre otras formas.

Por ejemplo, cuando un foro de internet pide el “Login”, es porque está pidiendo el nombre de usuario y contraseña que se eligió cuando el usuario se registró en dicho foro de internet.

El Login suele usarse como verbo y conjugarse al españolizarse, por ejemplo: “loguearse”. En inglés, la acción “loguearse” es “login”.

Un término más apropiado para “loguearse” sería “Iniciar sesión” o “Autenticarse”.

En proceso de Login es esencial para la seguridad y privacidad en internet que permite que solamente el usuario autorizado tenga acceso a los servicios y sistemas por medio de un método de seguridad, normalmente con usuario y contraseña. Es por eso que es importante elegir contraseñas seguras y cambiarlas regularmente, así como también no compartirlas con terceros o utilizarlas en sitios no confiables.

En resumen, el login es el procedimiento de autenticación mediante el cual un servicio accede a un servicio o sistema a través de un nombre de usuario y contraseña u otras formas de identificación biométrica. Es una herramienta fundamental para la seguridad y privacidad en la red, como lo dice la referencia (ALEGSA.com.ar, 2022).

3.1.5.5. Elementos para crear el Login

Los elementos que se necesitan para crear el Login con C# son:

Form

Label(etiqueta).

Textbox (caja de texto).

Botón.

Los elementos que se mencionan se explicarán de manera ordenada, como se va realizando el proyecto.

3.1.5.6 Form

Form: Form es la ventana que aparece en el momento de acceder a un nuevo proyecto C# con Windows Forms.NET Framework como se muestra en la figura 18, y en la figura 19 se observa la ventana form1 que aparece cuando seleccionamos el proyecto de C# (Windows Forms.NET Framework).

Figura 18. Nuevo proyecto Aplicación de Windows Forms(NET Framework).

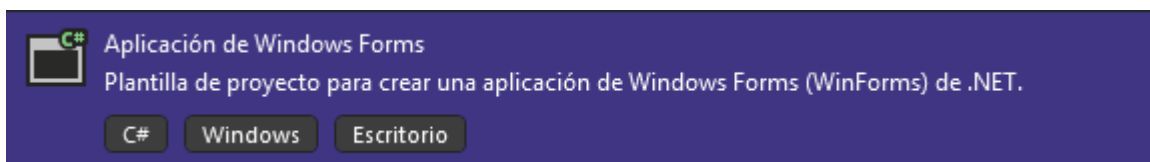
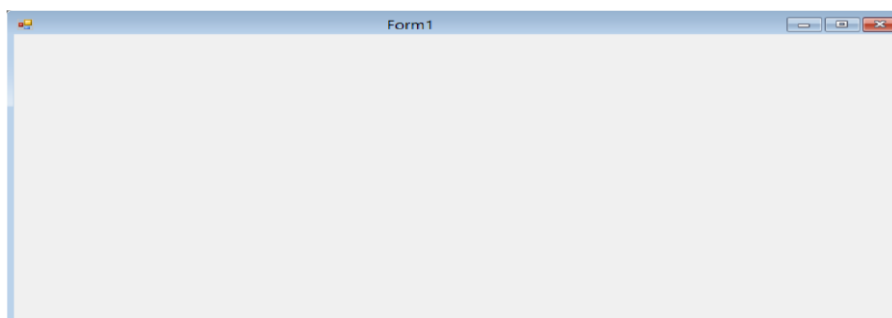


Figura 19. Form1 del proyecto Windows Forms NET Framework.



3.1.5.7 . Uso de elementos Label (etiqueta) para títulos de datos.

Label: La traducción en inglés es etiqueta, el elemento Label permite mostrar básicamente un texto que se modifica con la propiedad Text.

En el proyecto el elemento Label se requiere para colocar un título que sirve para identificar el dato que se necesita introducir, en este caso en el login se necesita el título de usuario y el título de contraseña, en la figura 20 se muestra un ejemplo de label(etiqueta) sin agregar algún título.

Nota: El elemento label (etiqueta) en el momento de arrastrar a Form, ya está incluido el título label.

Figura 20. Label1, en español etiqueta elemento para mostrar básicamente un título.



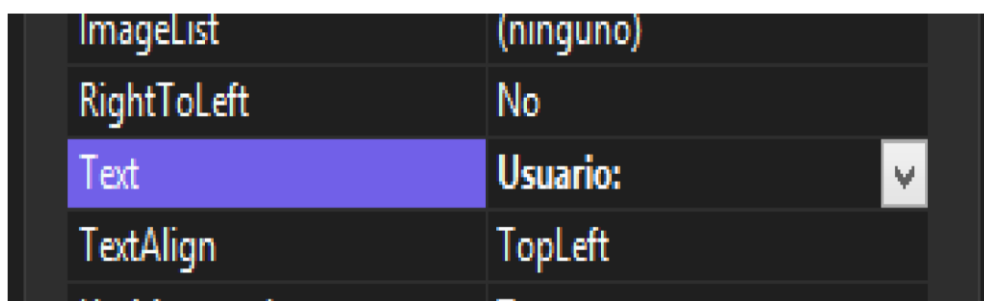
Para todo perfil normalmente se necesita un usuario y contraseña, debe haber una forma de agregar los títulos de los datos, en este caso se requieren elementos label(etiqueta).

Los datos que se necesitan identificar en el login son Usuario y Contraseña para agregar correctamente los datos que se necesitan, en el login y en todas las ventanas que se utilizará elemento label (etiqueta), no existirá programación por el momento, únicamente asignación de títulos.

Usuario: Aquella persona que habitualmente se introduce a un sistema para actualizar lo que se necesita, en el login de este proyecto el usuario necesita acceder al sistema para registrar algún trabajador nuevo.

Para comenzar con el procedimiento de agregar títulos, arrastramos los elementos label, una vez que se tiene arrastrado el elemento label (etiqueta), se presiona clic derecho y se accede a propiedades, para agregar texto (text), la propiedad Text es prácticamente el título del dato que nos dice que se necesita introducir y se mostrará de manera visible, en la figura 21 se muestra el título del primer dato que se introduce que es el título Usuario:

Figura 21. Texto de Label asignado el nombre de usuario.



Contraseña: Es una clave que permite acceder a un lugar, ya sea en el mundo real o en el virtual, se utiliza por varios motivos.

Preservar la intimidad, mantener uso en secreto, como una medida de seguridad, o como una combinación de todo ello.

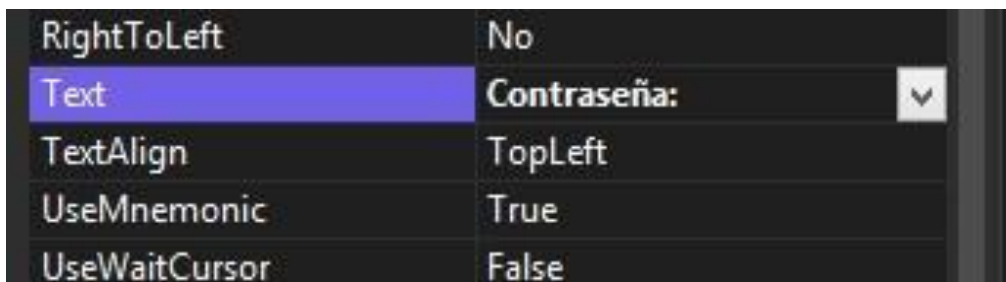
Tener estricta confidencialidad de cualquier proyecto.

Evitar cualquier plagio de proyecto.

Prevenir cambios innecesarios en el proyecto, en caso de que acceda intente acceder al sistema una persona desconocida.

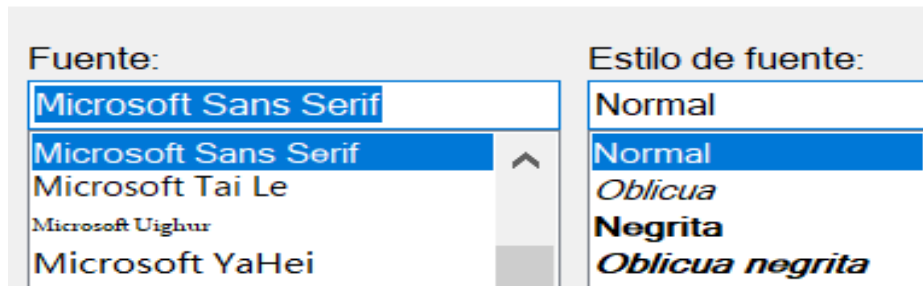
Para agregar el título del segundo dato que es la contraseña, se arrastra o copea el label (etiqueta) de usuario, pero para no confundir, de manera normal se arrastra otro elemento label (etiqueta), posteriormente se presiona clic derecho y se escribe la palabra Contraseña: adecuadamente se escribió el siguiente título del dato Contraseña: en la figura 22 se muestra la agregación de título del dato contraseña:

Figura 22. Texto de Label asignando el nombre contraseña.



Para tener un tipo de letra se accede a la propiedad Font en la opción tipo de letra, para tener un título adecuado, no es obligatorio un tipo de letra, pero es proyecto profesional, debe tener una cierta formalidad, un proyecto de Ingeniero, en la figura 23 se muestra el tipo de letra que se utilizó para ambos datos Usuario y contraseña.

Figura 23. Tipo de letra para títulos de datos, Usuario y Contraseña.



Para modificar el tamaño de letra se presiona clic derecho y nuevamente se utiliza la opción de Font, para modificar de manera libre el tamaño de letra de los títulos Usuario: y contraseña: en la figura 24 se observa la propiedad Font y en la figura 25 se muestra el tipo de letra que se utilizó para ambos datos de Login Usuario: y Contraseña:

Figura 24. Despliegue de elemento Font.

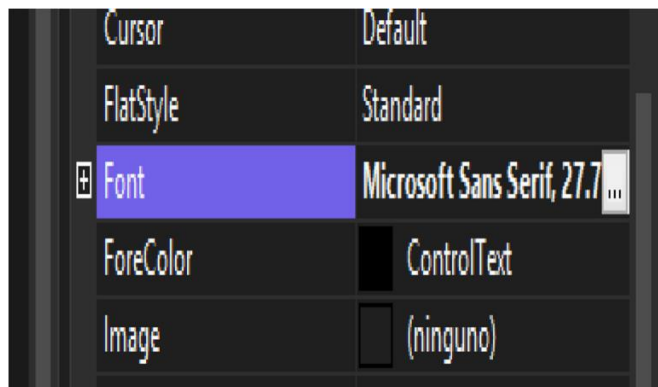
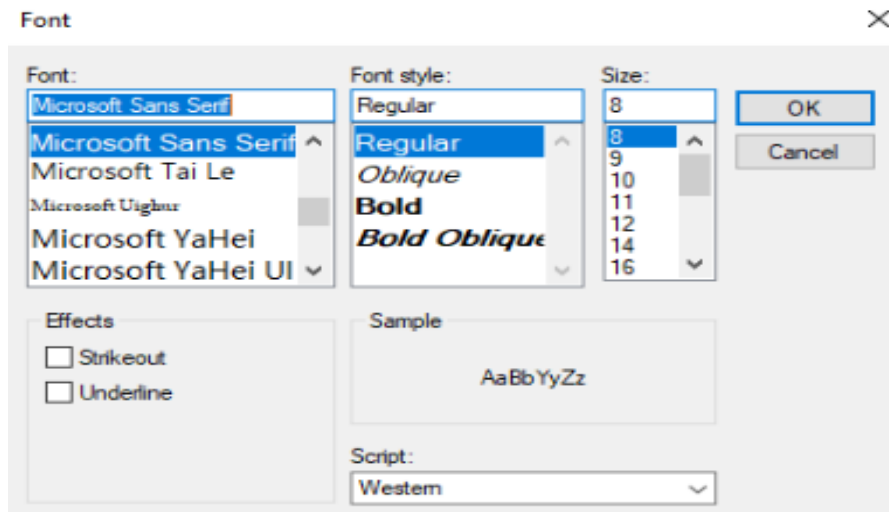


Figura 25. Opciones para cambiar tamaño de letra para Usuario y Contraseña.



3.1.5.8. Uso de textbox (caja de texto) para introducir y programar datos.

Para introducir cualquier dato, se necesitan los textbox (caja de texto), que son los elementos que utilizaremos en el login para introducir los datos usuario y contraseña para acceder.

Textbox (caja de texto): Permite agregar cualquier dato compuesto por letras y/o números que se asignan como únicos, en este caso el proyecto debe tener dos datos únicos de entrada que son usuario y contraseña. En la figura 26 se muestran los textbox que se necesitan.

Figura 26. Elemento textbox (caja de texto).



A screenshot of a login form on a light gray background. It contains two text input fields. The first field is labeled "Usuario:" and the second is labeled "Contraseña:". Both fields are empty and have a white background with a thin gray border.

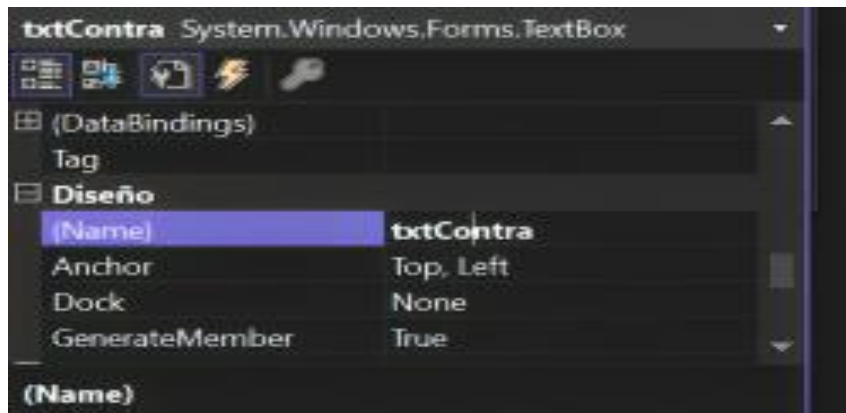
En la figura 26, se muestran cómo se acomodan adecuadamente los textbox (caja de texto), de forma alineada que ambos cumplan el mismo tamaño y no se observen de una forma desordenada. Para la adecuada programación de los textbox1 y textbox2 se debe asignar un nombre a cada textbox, para ubicar cada elemento en el momento de programar y tener una lógica correcta, en este caso solo se necesita agregar un nombre haciendo click derecho y accediendo a la propiedad Name, en este caso el nombre que se le asigna al textbox1 es Nombre, en la figura 27 se observa el cambio de nombre a textbox1.

Figura 27. Cambio de nombre textbox1 a Nombre



En la figura 28 se observa el cambio de nombre a textbox2, el nombre que se asignó fue txtContra.

Figura 28. Cambio de nombre textbox2 txtContra



3.1.5.9 Uso de botones de Windows Forms C#

Botón C#: El control Button de Windows Forms permite al usuario hacer clic en él para llevar a cabo una acción. Al hacer clic en el botón, parece como si se hubiera presionado y soltado. Cada vez que el usuario hace clic en un botón, se invoca el controlador de eventos Clic.

Para ingresar a cualquier base de datos, o página se necesita de un botón, en este caso el botón es botón virtual, pero se configurará para tener un funcionamiento requerido.

Se agregan 2 botones arrastrando hacia Form1, para programar adecuadamente las funciones que se necesitan realizar, la opción de iniciar y la opción de salir.

En la figura 29 se muestran los botones que se utilizarán para iniciar sesión y para salir. Nota: El botón de c# en el momento que se arrastra, automáticamente aparece el nombre de button 1, se deben cambiar accediendo a propiedades y con la función de text se agrega el título.

Figura 29. Ejemplo de botones agregados a login.

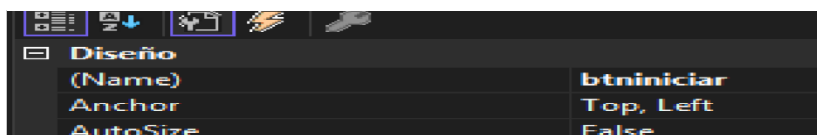


Asignación de nombre y texto para textbox (caja de texto):

Para determinar la función de los textbox en la programación, se asigna un nombre único para programar de manera adecuada y no tener nombres repetidos que puedan obtener una lógica errónea en el momento de programar.

Para tener una lógica correcta en el momento de presionar los botones, ya sea de iniciar sesión o salir, se presiona clic derecho en el textbox 1 para acceder a propiedades y por medio de la propiedad Name, se asigna el nombre que se desea utilizar, en este caso se asignó el nombre de btniniciar, en la figura 30 se observa el cambio de nombre de button 1 a btniniciar.

Figura 30. Nombre de botón iniciar.



Ahora en el botón se necesita un nombre visible, para saber que función estamos haciendo, la función del botón que se necesita es para iniciar sesión, para tener un nombre visible dentro del botón se realiza lo siguiente, presionar clic derecho para acceder a

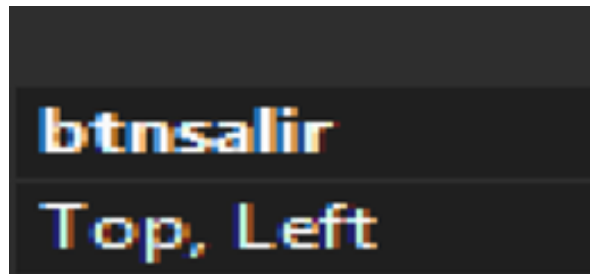
propiedades, y en la propiedad text se accede a cambiar de button1 a iniciar sesión en la figura 31 se observa el cambio de nombre en la propiedad a través de la propiedad text.

Figura 31. Cambio de botón 1 a iniciar sesión.



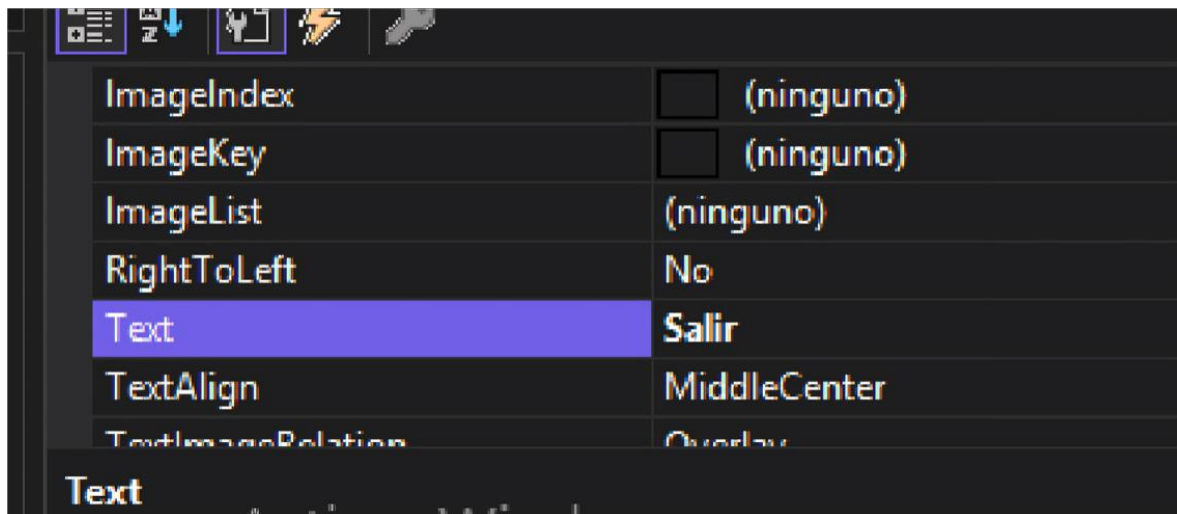
Por cualquier razón se necesita salir del login, se requiere de un botón, necesitamos asignar el nombre btnsalir que se asigna respectivamente, el nombre que se le asignó fue btnsalir para tener una lógica sin errores, en la figura 32 se observa el cambio de button 2 a btnsalir.

Figura 32. Nombre de botón Salir.



Ahora en el botón 2 se asigna un nombre visible, en este botón se presiona click derecho para acceder a propiedades, en la propiedad Text, se cambia de button 2, al nombre de salir, como se observa en la figura 33.

Figura 33. Ejemplo de cambio de título.



En la figura 34 se aprecia el login de una manera adecuada y ordenada, por el momento corriendo sin ningún tipo de programación en los botones.

Figura 34. Asignación de nombre de los botones.

Usuario:

Contraseña:

Algunas ideas para crear lo mostrado se obtuvieron de la referencia (ADVANCE, 2022).

4. Desarrollo de base de datos, y conexión (Interfaz de usuarios).

4.1 Inicio creación de la base de datos.

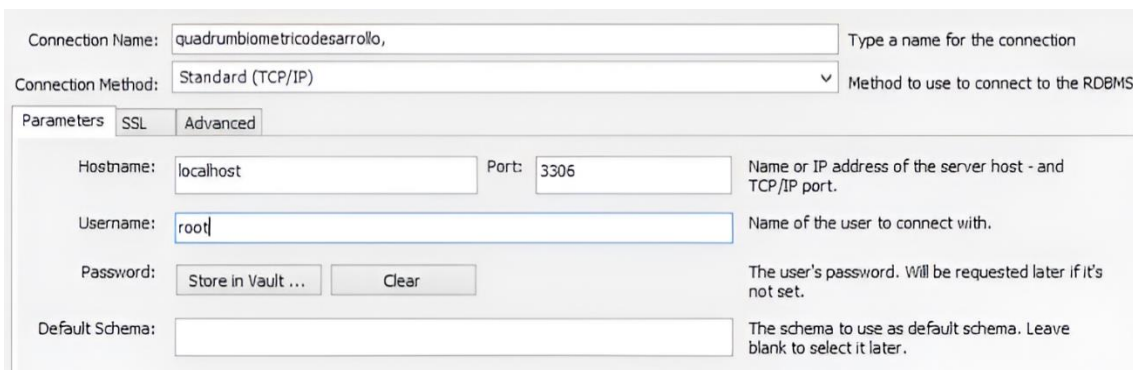
1.- Presionar el botón de (+) para desplegar la configuración de base de datos requerida, el botón se observa en la figura 35.

Figura 35. Opciones para crear base de datos con Gestor MySQL.



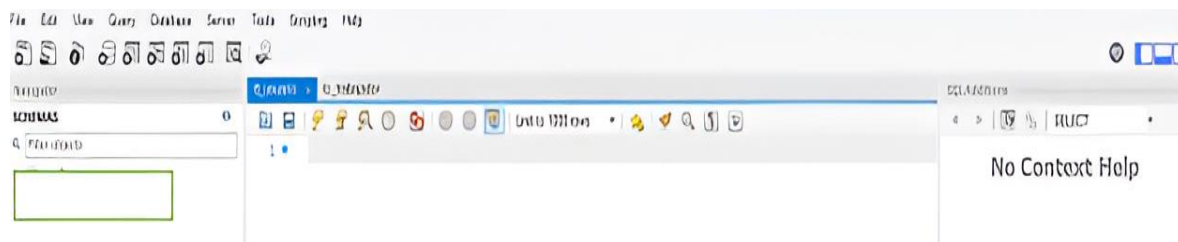
2.- Se agregan las configuraciones correspondientes de la conexión de la base de datos, el nombre de la base de datos se llamará quadrumbiometricodesarrollo, la conexión es local, por lo tanto, se llamará “localhost”, el usuario se llama root, con el puerto que trae establecido de número 3306 y la contraseña únicamente será admin, la configuración de la conexión de la base de datos se muestra en la figura 36.

Figura 36. Configuración de conexión a la base de datos

The screenshot shows a web form for configuring a MySQL connection. At the top, there is a text input for 'Connection Name' with the value 'quadrumbiometricodesarrollo' and a placeholder 'Type a name for the connection'. Below it is a dropdown menu for 'Connection Method' set to 'Standard (TCP/IP)' with a placeholder 'Method to use to connect to the RDBMS'. There are three tabs: 'Parameters', 'SSL', and 'Advanced', with 'Parameters' selected. The 'Parameters' section contains: 'Hostname' set to 'localhost', 'Port' set to '3306', 'Username' set to 'root', and 'Password' with a 'Store in Vault ...' button and a 'Clear' button. A 'Default Schema' field is empty. Each field has a descriptive tooltip on the right.

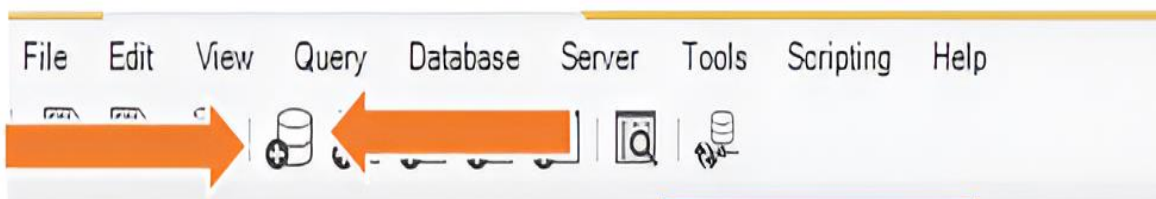
3.- Una vez accediendo a la conexión a la base de datos, se visualiza de la siguiente forma como se muestra en la figura 37.

Figura 37. Visualización de la base de datos en blanco.



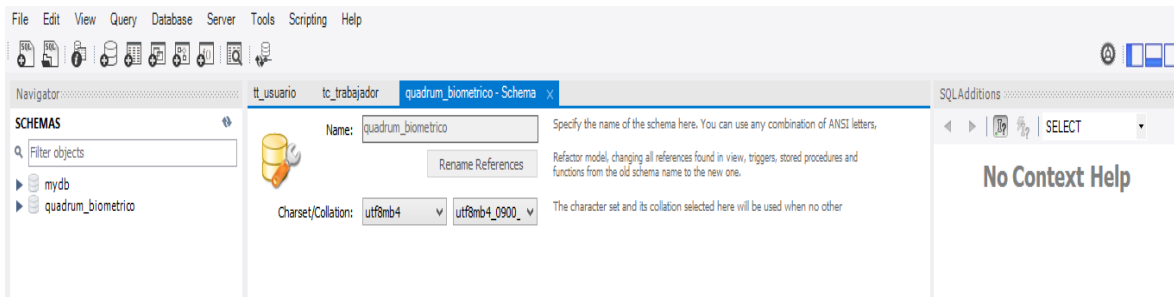
4.- Ahora se presiona el botón de la opción de crear base de datos como se aprecia en la figura 38.

Figura 38. Botón Creación base de datos.



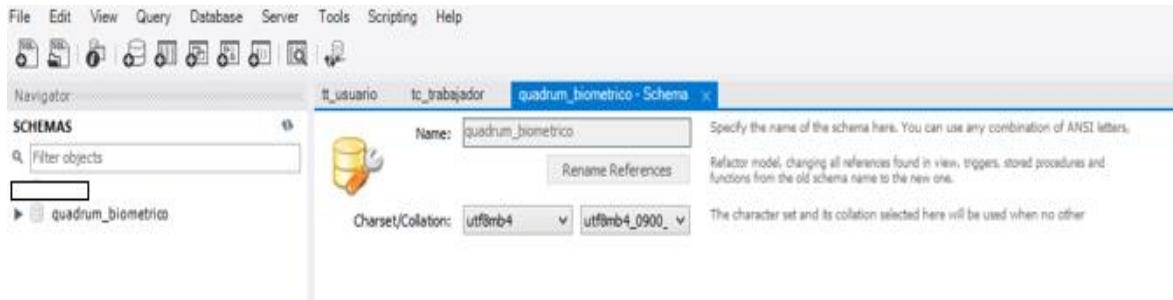
Como se puede apreciar en la figura 39 se muestra la opción de colocar nombre a la base de datos como se aprecia en la figura 40.

Figura 39. Visualización del nombre de la base de datos.



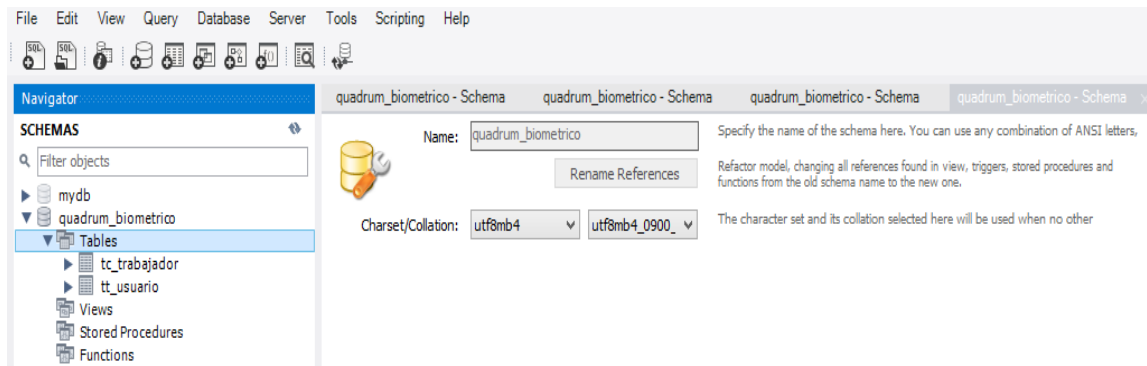
5.- Listo, en la figura 40 se aprecia que ya se creó la base de datos correspondiente.

Figura 40. Base de datos correspondientes.



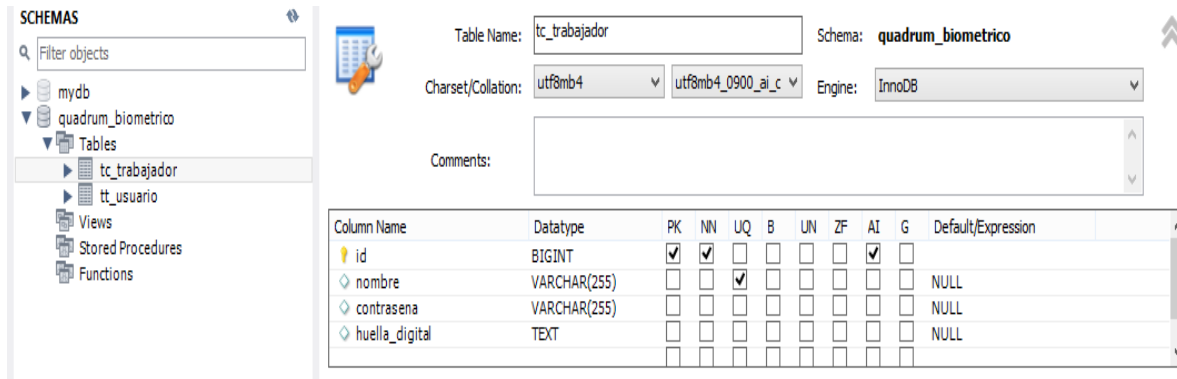
6.- Se crean las tablas correspondientes como se muestra en la figura 41.

Figura 41. Tablas correspondientes.



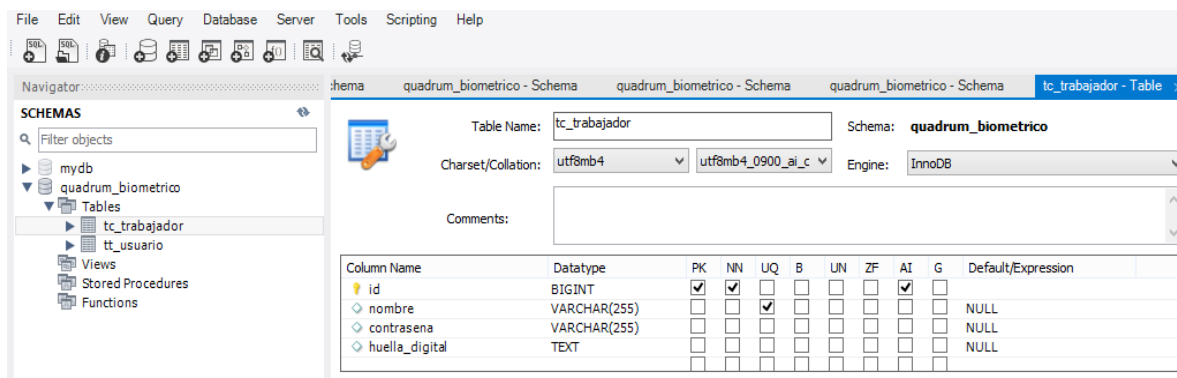
7.- Para la tabla tc_trabajador se agregan los datos id, nombre, contraseña, huella digital con sus configuraciones correspondientes, se visualiza en la figura 42.

Figura 42. Datos correspondientes de la tabla tc_trabajador.



8.-En la figura 43 muestran las configuraciones de tt_usuario con sus datos correspondientes y listo ya se tiene la base de datos.

Figura 43. Datos de tabla tt_usuario.



Es de sería importancia tener la base de datos configurada adecuadamente, para que más adelante no se generen errores, la forma para realizar una base de datos se obtuvo de la referencia (GILFILLAN, 2020).

4.2 CRD

Prácticamente en este proyecto se necesita hacer un Crud, que se define en inglés y traducido al español de la siguiente forma:

Create - Crear

Read- Leer.

Update - Actualizar

Delete - Eliminar.

Cuando se habla de Crate (Crear) es ingresar datos para guardar, en este caso en el proyecto lo que se debe realizar es una inserción de datos de trabajadores de la empresa que deben ser guardados y los datos son los siguientes:

1. Id
2. Nombre
3. Contraseña.
4. Huella digital.

4.2.1. Read- Leer - Consultar

Normalmente cuando se crea un CRUD en un proyecto profesional de Programación se empieza por partes, la primera parte es aplicar el método Read- Leer, que esto es realizar

una consulta de lo que se encuentra en la base de datos o en cualquier dispositivo que sea capaz de guardar, en este caso el dispositivo Biométrico.

El método consulta de trabajadores es un método para consultar los trabajadores registrados en el biométrico, y en este proyecto en la base de datos a través de un método de programación public void consultarTrabajadoresBiometrico ().

Para configurar la programación de consultarTrabajadoresBiometrico (), debe existir un **DataGridView** con las columnas correspondientes en el momento de consultar los trabajadores, se muestran las líneas de código del Método de consultar trabajadores, por ahora solo desarrollo del código.

Se realiza una consulta a la base de datos de MySQL para obtener información sobre los trabajadores desde la tabla 'tc_trabajador', luego, muestra los resultados en un control 'DataGridView' llamado 'ListaTrab'.

```
public void consultarTrabajadoresBiometrico()
{
    Console.WriteLine("llama consultarTrabajadoresBiometrico");

    MySqlConnection connection = ConexionMysql.conexionMysql();

    MySqlCommand com = connection.CreateCommand();

    com.CommandType = System.Data.CommandType.Text;

    com.CommandText = "SELECT * From tc_trabajador";

    MySql.Data.MySqlClient.MySqlDataReader reader =
com.ExecuteReader();

    if (reader.HasRows)
```

```

    {
        while (reader.Read())
        {
            int n = ListaTrab.Rows.Add();

            ListaTrab.Rows[n].Cells[0].Value = reader["id"].ToString();

            ListaTrab.Rows[n].Cells[1].Value = reader["nombre"].ToString();

            ListaTrab.Rows[n].Cells[2].Value = reader["contrasena"].ToString();

            ListaTrab.Rows[n].Cells[3].Value = "privilegio";

        }
    }
}

```

Establecer la Conexión a la base de datos.

```
MySqlConnection connection = ConexionMysql.conexionMysql();
```

Se establece una conexión a la base de datos de MySQL utilizando el método ‘conexionMysql’ de la clase ‘ConexionMysql’.

3.2.2 . Crear y Ejecutar la Consulta SQL.

```

    MySqlConnection connection = ConexionMysql.conexionMysql();
    MySqlCommand com = connection.CreateCommand(); com.CommandType =
    System.Data.CommandType.Text; com.CommandText = "SELECT * From tc_trabajador";
    MySql.Data.MySqlClient.MySqlDataReader reader = com.ExecuteReader();

```

Se crea un comando ('com') asociado con la conexión a la base de datos y se define una consulta SQL para seleccionar todos los campos de la tabla 'tc_trabajador'. La consulta se ejecuta, mediante el método 'ExecuteReader', y todos los resultados se almacenan en un lector de datos ('reader').

Procesar Resultados y Llenar el DataGridView

```
if(reader.HasRows)
{
while(reader.Read())
{
Int n = ListaTrab.Rows.Add();

ListaTrab.Rows[n].Cells[0].Value = reader["id"].ToString();

ListaTrab.Rows[n].Cells[1].Value = reader["nombre"].ToString();

ListaTrab.Rows[n].Cells[2].Value = reader["contrasena"].ToString();

ListaTrab.Rows[n].Cells[3].Value = reader["privilegio"].ToString();

}

}
```

Se verifica si el lector de datos tiene filas. Si hay filas, se utiliza un bucle **while** para leer cada fila y agregar una nueva fila al **DataGridView** (**ListaTrab**) por cada resultado. Los valores de las columnas se obtienen del lector de datos y se asignan a las celdas correspondientes en el **DataGridView**.

Es importante destacar que en la celda correspondiente al privilegio (ListaTrab.Rows[n].Cells[3].Value), se asigna el valor constante "privilegio" en lugar de obtenerlo del lector de datos. Si el valor del privilegio está almacenado en la base de datos, deberías usar el campo correspondiente del lector de datos en lugar de la cadena constante.

Finalmente, este código básicamente recupera información de la base de datos y la muestra en un **DataGridView** en tu aplicación. Si tienes más preguntas o necesitas aclaraciones sobre algún aspecto específico, no dudes en preguntar, en la tabla 1 se muestra un dataGridView que por el momento se encuentra sin columnas de la base de datos.

Tabla 1. Ejemplo de Datagridview, para la base de datos.

Nombre	Apellidos	Edad
Javier	Aguilar	25

En la tabla 2 se muestra la tabla de datagirview, junto con los datos correspondientes de la base de datos, Por ahora se muestra únicamente con las puras columnas de la base de datos funcionar.

Tabla 2. Datagridview con las columnas de la base de daos.

Número de Registro	Nombre de Trabajador	Contraseña	ID Privilegio	Agregar Huella
--------------------	----------------------	------------	---------------	----------------

3.2.3. Configuración de datos de base de datos.

Para la configuración de acceso al software se cambia la forma de declarar el usuario y contraseña, todo esto se realiza con la ayuda de la base de datos, porque es una forma segura de tener datos delicados, en este caso el usuario y contraseña de quien es encargado de agregar nuevos trabajadores y eliminar los trabajadores que renuncian, o normalmente son despedidos por cualquier circunstancia, en la tabla 3 se muestra el usuario y contraseña del usuario administrador creado en la base de datos Mysql.

Tabla 3. Usuario y contraseña de administrador.

id	usuario	contrasena	estado
1	admin	1234	administrador

Por los cambios que se realizan en el login con la interacción a la base de datos de Mysql se modifica la función de `btniniciar_Click(object sender, EventArgs e)`, que ahora el usuario y contraseña se almacenan en la base de datos de MySQL, todo esto relacionado con el login.

1.- Se establece una conexión a la base de datos MySQL utilizando el método **conexión Mysql** de la clase **Conexión Mysql**.

```
MySqlConnection connection = ConexionMysql.conexionMysql();
```

2.- Crear y Ejecutar la Consulta SQL.

```
MySqlCommand com = connection.CreateCommand();
```

```
com.CommandType = System.Data.CommandType.Text;
```

```
com.CommandText = "SELECT * From tt_usuario WHERE usuario=" + "" + Nombre.Text + "" + "AND contrasena=" + "" + txtContra.Text + "";
```

```
        MySql.Data.MySqlClient.MySqlDataReader        reader        =  
com.ExecuteReader();
```

Se crea un comando (**com**) asociado con la conexión a la base de datos y se define una consulta SQL para seleccionar todos los campos de la tabla **tt_usuario** donde el nombre de usuario (**usuario**) coincide con el valor en el cuadro de texto **Nombre** y la contraseña (**contraena**) coincide con el valor en el cuadro de texto **txtContra**. La consulta se ejecuta mediante el método **ExecuteReader**, y los resultados se almacenan en un lector de datos (**reader**).

3.- Verificar Existencia de filas resultantes.

```
if (reader.HasRows)  
  
    {  
  
        this.Hide();  
  
        MenuBiometrico NuevaVenta = new MenuBiometrico();  
  
        NuevaVenta.Show();  
  
        reader.Close();  
  
    }
```

Se verifica si el lector de datos tiene filas. Si hay filas, significa que se encontró un usuario con los datos apropiados. En este caso, se oculta el formulario actual, se crea un nuevo formulario **Menú Biométrico** y se muestra. Luego, se cierra el lector de datos.

4.- Verificación de contraseña incorrecta.

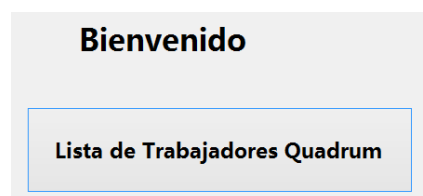
```
else
```

```
{  
  
    MessageBox.Show("Intente de nuevo por favor");  
  
    Nombre.Text = "";  
  
    txtContra.Text = "";  
  
    Nombre.Focus();  
  
}
```

Si no hay filas en el resultado, se muestra un mensaje de `MessageBox` indicando al usuario que las credenciales son incorrectas. Además, se limpian los cuadros de texto **Nombre** y **txtContra**, y se establece el enfoque en el cuadro de texto **Nombre** para facilitar un nuevo intento.

Listo, una vez correctos los datos se accede directamente a la nueva ventana en este caso el menú, como se muestra en la siguiente figura 44.

Figura 44. Menú de Bienvenido.



Una vez que se accede al menú, se presiona el botón de lista de trabajadores para realizar mi consulta del contenido en la base de datos creada, tal como se muestra en la tabla 4 también se puede observar que por el momento se encuentra vacía la tabla `Datagridview`.

Tabla 4. `Datagridview` de consulta de trabajadores.

Número de Registro	Nombre de trabajador	Contraseña	Id Privilegio	Agregar Huella

3.2.4. Realización de insert con la base de datos y el Biométrico.

Para la realización del insert se comienza de la siguiente manera:

1.- Agregar un nuevo formulario, con sus respectivos componentes que son las etiquetas, los textbox y los botones. como se muestra en la figura 45.

Figura 45. Nuevo formulario para nuevo método insertar.

Ahora se debe programar el botón de etiqueta agregar para tener la inserción a la base de datos correspondiente, una vez que se tienen los datos y los nombres asignados de los componentes, se comienza con el botón de etiqueta Agregar.

La programación del botón con la etiqueta de agregar se realiza de la forma presionando clic dos veces para desplegar la función del botón correspondiente, como se muestra en el siguiente ejemplo de líneas de código.

```
private void button1_Click(object sender, EventArgs e)
{
    try
    {
        if (String.IsNullOrEmpty(this.Nombre.Text))
            throw new Exception("Nombre trabajador Requerido");

        if (String.IsNullOrEmpty(this.txtContra.Text))
            throw new Exception("Contraseña requerida");

        MySqlConnection conexionMysql = ConexionMysql.conexionMysql();

        string sql = "INSERT INTO tc_trabajador(nombre,contrasena )
VALUES(@nombre, @contrasena)";

        MySqlCommand comando = new MySqlCommand(sql,
conexionMysql);

        comando.Parameters.AddWithValue("@nombre", Nombre.Text);

        comando.Parameters.AddWithValue("@contrasena", txtContra.Text);

        int resultado=comando.ExecuteNonQuery();

        MySqlCommand com = conexionMysql.CreateCommand();

        com.CommandType = System.Data.CommandType.Text;
```

```
com.CommandText = "SELECT * From tc_trabajador WHERE  
nombre=" + "\"" + Nombre.Text + "\"";
```

```
MySQL.Data.MySqlClient.MySqlDataReader reader =  
com.ExecuteReader();
```

```
if (reader.HasRows) {
```

```
while (reader.Read()) {
```

```
Console.WriteLine(reader["id"].ToString());
```

```
Console.WriteLine(reader["nombre"].ToString());
```

```
Console.WriteLine(reader["contrasena"].ToString());
```

```
agregarUsuarioBiometrico(reader["id"].ToString(),  
reader["nombre"].ToString(), reader["contrasena"].ToString(), 0);
```

```
this.Nombre.Text = "";
```

```
this.txtContra.Text = "";
```

```
this.Hide();
```

```
}
```

```
}
```

Ahora como se observa el nombre de la función de nombre Button1_Click, la clase es privada, y ahora se explica lo siguiente.

Como primer paso se accede a realizar la inserción con la base de datos, para más adelante desarrollar la función con el biométrico.

```
private void button1_Click (object sender, EventArgs e)
```

Este código se ejecuta cuando se hace clic en el button 1, como ya se ha explicado anteriormente, ahora se muestra una validación de entrada que verifica si el Nombre está vacío o nulo.

This.Nombre. Text, obtiene el texto ingresado en el control de texto llamado Nombre, si el campo está vacío o nulo se lanza una excepción con el mensaje de Exception que se captura con la línea de código throw new Exception “Nombre trabajador requerido.”

```
if (String.IsNullOrEmpty(this.Nombre.Text))  
  
    throw new Exception("Nombre trabajador Requerido");
```

Ahora se muestra la validación que verifica si el campo txtContra que es el nombre que se le asigna al inputText de contraseña, esto quiere decir que cuando se presiona el botón de agregar sin el dato de txtContra lanza la excepción de contraseña requerida.

```
if (String.IsNullOrEmpty(this.txtContra.Text))  
  
    throw new Exception("Contraseña requerida");
```

El nombre es el input Text que se le asigna en las configuraciones de los componentes, como se muestra en la figura 46 el textBox de Nombre. y en la figura 47 se muestra el textBox de txtContra.

Figura 46. Textbox con el nombre de Nombre.

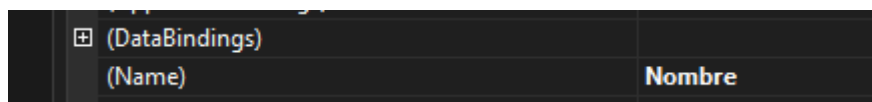
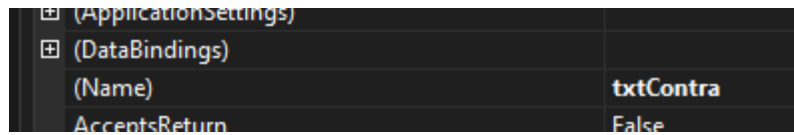


Figura 47. TextBox con el nombre de txtContra.



Los anteriores nombres los buscará la validación de insertar datos, si no encuentra esos dos nombres, se obtiene un error.

Ahora se explican las siguientes líneas de código.

```
MySqlConnection conexionMysql = ConexionMysql.conexionMysql();

        string sql = "INSERT INTO tc_trabajador(nombre,contrasena )
VALUES(@nombre, @contrasena)";

        MySqlCommand comando = new MySqlCommand(sql,
conexionMysql);

        comando.Parameters.AddWithValue("@nombre", Nombre.Text);

        comando.Parameters.AddWithValue("@contrasena", txtContra.Text);

        int resultado=comando.ExecuteNonQuery();

        MySqlCommand com = conexionMysql.CreateCommand();

        com.CommandType = System.Data.CommandType.Text;

        com.CommandText = "SELECT * From tc_trabajador WHERE
nombre=" + "\"" + Nombre.Text + "\"";

        MySql.Data.MySqlClient.MySqlDataReader reader =
com.ExecuteReader();, a continuación se muestra línea por línea del código.
```

```
MySqlConnection conexionMysql = ConexionMysql.conexionMysql();
```

Se crea la instancia de MySqlConnection llamada conexión Mysql.

Se utiliza el método `conexionMysql` de la clase `ConexionMysql` para obtener y establecer la conexión con la base de datos.

La siguiente línea de código es la definición de datos definido string sql

```
string sql = "INSERT INTO tc_trabajador(nombre,contrasena )  
VALUES(@nombre, @contrasena)";
```

Se define una cadena 'sql' que representa la consulta de la inserción en la tabla `tc_trabajador`.

La consulta utiliza los parámetros `@nombre` y `contraseña` para evitar posibles problemas de seguridad como la inyección de SQL,

```
string sql = "INSERT INTO tc_trabajador(nombre,contrasena )  
VALUES(@nombre, @contrasena)";
```

la cadena sql se utiliza en el gestor de base de datos, para mandar lo que es la inserción del nuevo trabajador., pero en esta línea de código se implementó para tener el resultado de agregar el nuevo trabajador.

La siguiente línea de código se necesita para la creación de un Comando SQL.

```
MySqlCommand comando = new MySqlCommand(sql, conexionMysql);
```

Se crea el objeto `MySqlCommand` llamado 'comando', asociado a la consulta 'sql' y la conexión 'conexionMysql'.

Las siguientes dos líneas de código asignan parámetros.

```
comando.Parameters.AddWithValue("@nombre", Nombre.Text);
```

```
comando.Parameters.AddWithValue("@contrasena", txtContra.Text);
```

Se asignan valores a los parámetros @nombre y @contrasena del comando, los valores provienen de los elementos de texto (textbox) 'Nombre' y 'txtContra'.

Ahora con la siguiente línea se ejecuta la consulta de inserción en la base de datos usando "comando.ExecuteNonQuery()".

.ExecuteNonQuery se utiliza para ejecutar consultas que no devuelven datos (como las consultas de actualización, inserción o eliminación).

El resultado indica el número de filas afectadas por la consulta.

Como prueba del funcionamiento del código de función llamado agregarUsuarioBiometrico, se ejecuta la función para guardar en la base de datos (private void button1_Click(object sender, EventArgs e)), como ya se menciona se necesita guardar en base de datos y en biométrico, ya se detalló la función para guardar en la base de datos, ahora se hace explicación para guardar en el biométrico, prácticamente funcionan al mismo tiempo al momento de guardar el nuevo trabajador.

```
private void agregarUsuarioBiometrico(string numeroRegistro, string nombre,
string contrasena, int privilegio).
```

```
{
    bool bEnabled = true;
    int iMachineNumber = 1;
    zkemkeeper.CZKEM axCZKEM1 = Utileria.coneccionBiometrico1();
    int idwErrorCode = 0;
    axCZKEM1.EnableDevice(iMachineNumber, false);
```

```

        if (axCZKEM1.SSR_SetUserInfo(iMachineNumber, numeroRegistro,
nombre, contraseña, privilegio, bEnabled))//upload the user's information(card number
included)

        {

            MessageBox.Show(String.Format("El trabajador {0} se registró en el
Biometrico", nombre));

        }

        else

        {

            axCZKEM1.GetLastError(ref idwErrorCode);

            throw new Exception(String.Format("Operation failed,ErrorCode= {0}
Error", idwErrorCode.ToString()));

        }

        axCZKEM1.RefreshData(iMachineNumber);//the data in the device should
be refreshed

        axCZKEM1.EnableDevice(iMachineNumber, true);

    }

```

private void agregarUsuarioBiometrico(string numeroRegistro, string nombre, string contraseña, int privilegio) Este método recibe cuatro parámetros ‘numeroRegistro’ (presumiblemente el identificador único). Nombre (nombre del trabajador), contraseña (la contraseña del trabajador), privilegio(nivel de privilegio de usuario).

Bool bEnabled= true;; Se declara y asigna un valor booleano ‘true’ a la variable bEnabled, Este indica que el usuario está habilitado.

Int iMachineNumber=1;” Se declara y asigna el valor ‘1’ a la variable iMachineNumber, es el número de identificación o dirección del dispositivo biométrico.

Zkemkeeper.CZKEM axCZKEM1= Utileria.coneccionBiometrico1()); “:Se utiliza el método coneccionBiometrico1 de la clase utilería de la clase Utilería para obtener una instancia de la clase CZKEM del dispositivo biométrico.

Int idwErrorCode=0; Se declara la variable ‘idErrorCode’ e inicializa con el valor ‘0’.Se utilizará para almacenar códigos de error en caso de que la operación falle.

AxCZKEM1.EnableDevice(iMachineNumber, false); Se deshabilita el dispositivo con el número iMachineNumber.

if (axCZKEM1.SSR_SetUserInfo(iMachineNumber, numeroRegistro, nombre, contrasena, privilegio, bEnabled)): Se intenta agregar la información del usuario al dispositivo biométrico se utiliza el método SSR_SetUserInfo. Si la operación tiene éxito, muestra un mensaje de éxito en un cuadro de mensaje.

Si se obtiene fallo en la operación, se obtiene el código de error utilizando axCZKEM1.GetLastError(ref idErrorCode): y se lanza una excepción con un mensaje que indica el código de error.

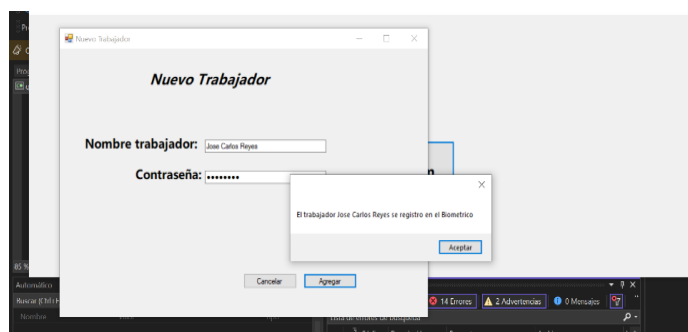
axCZKEM1.RefreshData(iMachineNumber); Refresca los datos en el dispositivo biométrico asociado al número iMachineNumber.

axCZKEM1.EnableDevice(iMachineNumber, true); “: Habilita nuevamente el dispositivo bioétrico.

El código se utiliza para agregar un usuario con información específica al dispositivo biométrico, con manejo de errores y actualización de datos en el dispositivo Biométrico MB160.

Una vez funcionando ambas funciones se arroja un mensaje satisfactorio con el que se muestra en la figura 48. Desplegado a través de una ventana de resultado satisfactorio.

Figura 48. Registro de usuario Biométrico.

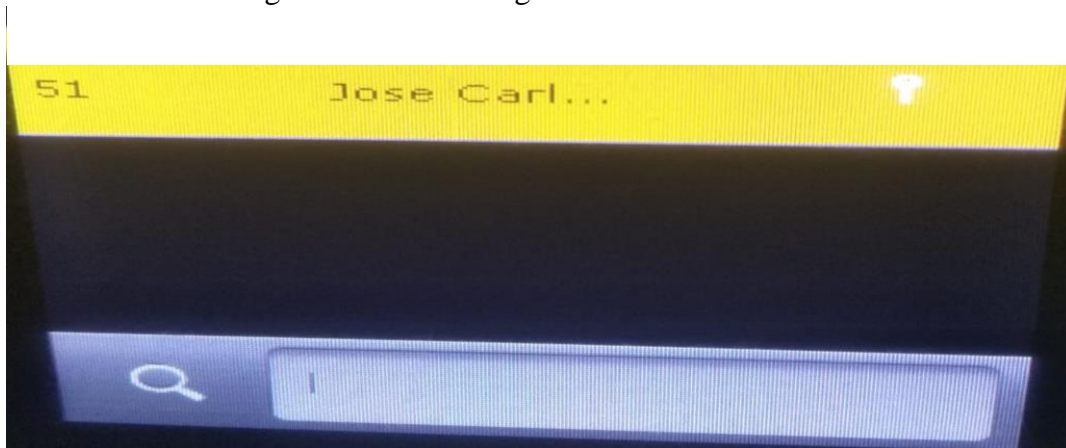


3.2.5. Verificación de existencia trabajadores en el biométrico.

En la figura 48 se muestra el mensaje que el usuario ya se registró en el biométrico.

Para comprobar que existe en el biométrico de manera manual se debe acceder a visualizar el biométrico de manera para comprobar que efectivamente se encuentra el registro del Nuevo trabajador, tal como se observa en la figura 49.

Figura 49 Usuario Registrado en el biométrico.



3.2.6.-Consulta de Usuario en DataGridView de C#.

Se puede verificar que el nuevo usuario se encuentra registrado dentro de la tabla realizando con sus respectivos datos por ahora establecidos, tal como se muestra en la tabla 5.

Tabla 5. Lista de registro de usuarios.

4	felo	234234234	2
12	jose	12312	3
122	marco	asfasdf	4
2123	lucas	asasd	5
123	rojo	qweqffg	6
12	libros	wqsdfsd	7
34	jesus	asas	8

Para tener todas las ideas que se muestran anteriormente se necesitó de la fuente (ZKT, 2022), ya que es una fuente oficial de los biométricos ZKT para vincular con C# y cualquier base de datos que requiera vinculación con los biométricos.

3. 2.7. Código para generar delete en c# y Mysql

Para eliminar un usuario por el momento por el momento se accede a la base de datos aplicando la respectiva función de lenguaje sql, primero se accede realizando un `SELECT * FROM` para realizar la consulta y verificar que el usuario existe, la línea completa es la siguiente `SELECT * FROM quadrum_biometrico.tc_trabajador;` como se observa en la tabla 6.

Tabla 6. Consulta del lado de base de datos MySQL.

SLECT * FROM quadrum_biometrico.tc_trabajador			
id	nombre	contrasena	Huella_digital
36	Felo	carrasco	

Posteriormente se consulta que exista del lado de la tabla de C#, y realizando la consulta si se encuentra el usuario, por el momento ambos datos coinciden como se muestra en la Tabla 7.

Tabla 7. Tabla de usuario encontrado.

Total de trabajadores.				
Numero de Registro	Nombre de trabajador	Contraseña	ID Privilegio	Regplicar Huella
36	Felo	Carrasco	privilegio	Replica Huella

Efectivamente se observa que, si existe el usuario, por lo tanto, los datos coinciden.

Ahora la necesidad que se tiene es eliminar el usuario, por lo tanto se accede aplicando la siguiente línea de sql `DELETE FROM quadrum_biometrico.tc_trabajador.`

`WHERE nombre = 'NombreEspecifico';` que eliminará los registros de la tabla `tc_trabajador` en la base de datos `quadrum_biometrico` donde el campo 'nombre' sea igual al `NombreEspecifico` que se solicita.

Una vez aplicado, nuevamente se aplica el `select * from` para revisar que el usuario se haya eliminado, una vez realizado el `select*from` se observa que efectivamente se ha eliminado el usuario del lado de la base de datos de MySQL como se observa en la tabla 8.

Tabla 8. Consulta de usuario eliminado.

SELECT * FROM quadrum_biometrico.tc_trabajador;			
id	nombre	contrasena	Huella_digital

Posteriormente se realiza la visualización del lado de la tabla de C# para observar que efectivamente está eliminado el usuario, y efectivamente el usuario está eliminado, tal como se observa en la tabla 9.

Tabla 9. Usuario eliminado al 100%.

Total de trabajadores					
Número de Registro	Nombre de trabajador	de	Contraseña	Id Privilegio	Replicar Huella

Esta función de delete únicamente se observa del lado de la tabla de C#, por el momento no es posible visualizar los resultados del lado del biométrico.

Todo este aprendizaje acerca de la base de datos se obtuvo de la referencia del libro (LUIS ALBERTO CASILLAS SANTILLÁN M. A., 2012), fue de mucha ayuda para obtener ideas de cómo vincular datos con C#, Mysql y el Biométrico ZKT (eco).

3. 2.8. Código para generar update en c# y Mysql

Para generar la función de update (actuaizar) se accede a realizar el select*from desde la base de datos para verificar que el usuario existe dentro de la base de datos, como se muestra en la tabla 10.

Tabla 10. Select*from de la base de datos para verificar que existe un usuario.

SELECT * FROM quadrum_biometrico.tc_trabajador			
id	nombre	contrasena	Huella_digital
37	felo carrasco comonfort	123	

Ahora que se observa que dentro de la base de datos existen datos de algún usuario se acceder a la tabla de C# para comprobar que este usuario coincide como se muestra en la tabla 11.

Tabla 11. Verificación de la tabla C# Revisión de usuario existente.

Total de trabajadores						
Numero de Registro	de	Nombre de trabajaor	de	Contraseña	ID Privilegio	Replicar Huella
37		felo carrasco comonfort		124	privilegio	Replicar Huella

Una vez que se observa que sí coinciden los datos del lado de la tabla de C# se accede a la tabla de la base de datos para acceder con la línea de código especialmente para modificar UPDATE quadrum_biometrico.tc_trabajador, efectivamente en la tabla 12. se observa que si funciona la línea de código sql UPDATE quadrum_biometrico.tc_trabajador.

SET nombre = 'Gabriel'

WHERE nombre = 'felo carrasco comonfort'; si funciona para actualizar.

Tabla 12. Tabla actualizada MySql.

Id	nombre	contrasena	Huella digital
37	Gabriel	1234	

Una vez que se actualiza del lado de la base de datos de MySQL se observa del lado de la tabla de C# para verificar que efectivamente se haya actualizado, como se observa en la tabla 13.

Tabla 13. Verificación de tabla actualizada tabla proyecto.

Total de trabajadores.				
Número de Registro	Nombre de trabajaor	Contraseña	ID Privilegio	Replicar Huella
37	Gabriel	123	orivilegio	Replicar Huella

Listo, una vez que se observa que si funciona se cierra el proceso de update, los métodos que se aplicaron se obtuvieron de la referencia (Cero, 2022).

4. Réplica de usuario en dos biométricos con datos correspondientes.

La réplica del segundo Biométrico es para registrar un trabajador en dos empresas, cuando llega un trabajador nuevo, se necesita que el trabajador sea registrado en la empresa “Centro de Validación digital” Av. J. M. Morelos Y P. 4 Ote. 1300, San Sebastián, 50090 Toluca de Lerdo, Méx, como principalmente registrar el usuario en la empresa principal ubicada en Montecito No. 38, Nápoles, Benito Juárez, 03810 Ciudad de México, CDMX, ambos biométricos deben tener registrados los usuarios en las dos empresas para tener su control de hora de entrada/salida/comida, todo esto es necesario porque en caso de que el trabajador sea requerido en la empresa de Toluca o de Ciudad de México se tenga su registro sin ningún problema.

Ahora se realiza la réplica de usuario, ¿Cómo se realiza?, se realiza utilizando la programación que ya se tiene asignada en la clase de Utilería creada con el lenguaje C# para un solo biométrico, únicamente es la creación de una función para la interacción con un solo biométrico, ya no tiene nada que ver con modificación de la base de datos porque se usa una sola para almacenar los datos de los usuarios.

Como primer paso se debe acceder a la clase utilería, donde se tiene la interacción con el hardware Biométrico1 para interactuar con el Hardware Biométrico2, son prácticamente las mismas líneas de código, únicamente crea un método función con diferente nombre, a continuación, se observa el método de programación para interacción con el Biométrico 1 y Biométrico 2.

```
public class Utileria
{
    //Método para Biometrico1

    public static zkemkeeper.CZKEM coneccionBiometrico1()
```

```

{
    zkemkeeper.CZKEM axCZKEM1 = new zkemkeeper.CZKEM();

    axCZKEM1.Connect_Net("192.168.100.63", 4370);

    return axCZKEM1;
}

```

//Metodo para Biometrico 2

```

public static zkemkeeper.CZKEM coneccionBiometrico2()
{
    zkemkeeper.CZKEM axCZKEM2 = new zkemkeeper.CZKEM();

    axCZKEM2.Connect_Net("192.168.100.64", 4370);

    return axCZKEM2;

    {

    }
}

```

En el ejemplo de líneas de código, se muestra la programación para la interacción con los Biométricos 1 y 2, Todo es Prácticamente lo mismo, lo único que cambia es el nombre del biométrico y la dirección IP.

Biométrico 1:

```

zkemkeeper.CZKEM axCZKEM1 = new zkemkeeper.CZKEM();//es el nombre de

```

```
axCZKEM1.Connect_Net("192.168.0.93", 4370); // Es la dirección Ip donde se va a conectar.
```

Biométrico 2.

```
zkemkeeper.CZKEM axCZKEM2 = new zkemkeeper.CZKEM(); // Nombre del Biométrico 2
```

```
axCZKEM2.Connect_Net("192.168.0.94", 4370); // Ip del Biométrico2.
```

Gracias al método de cambiar la dirección Ip se puede realizar con facilidad la interacción de los biométricos, para tener un registro.

El método para cambiar de dirección Ip se realiza para mandar datos a otro biométrico, se realiza por la razón que no deben coincidir las direcciones Ip, porque si coinciden, no será posible un funcionamiento adecuado, cada biométrico debe tener una dirección Ip diferente,

Por ejemplo:

Un usuario se encuentra conectado a la dirección Ip 192.168.0.109, si un tercero se conecta, no deben coincidir las direcciones IP en este caso el cuarto Octeto, porque si coinciden, no se tiene conexión de la red, ejemplo la otra persona debe estar conectada a la dirección 192.168.0.110.

Por lo tanto, los métodos de los biométricos del lado de la clase utilizaría tienen diferente dirección IP, cada método coincidiendo con las direcciones IP de los biométricos, a continuación, se muestra la línea de código donde se encuentra la dirección IP de cada biométrico y enseguida se muestran las figuras 50 del Biométrico 1 y figura 51 del Biométrico 2.

```
axCZKEM1.Connect_Net("192.168.100.63", 4370)
```

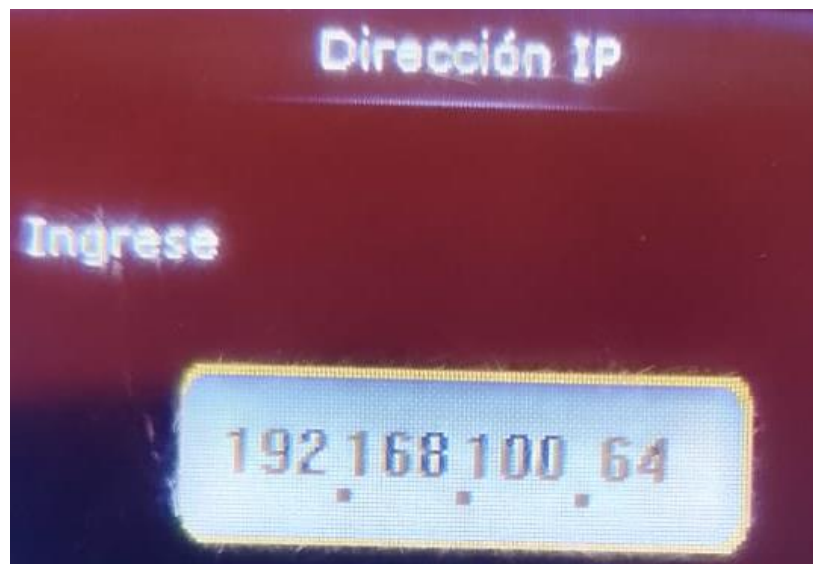
Figura 50. Dirección IP de Biométrico1.



Se observa la línea de código del Biométrico 2

```
axCZKEM2.Connect_Net("192.168.100.64", 4370);
```

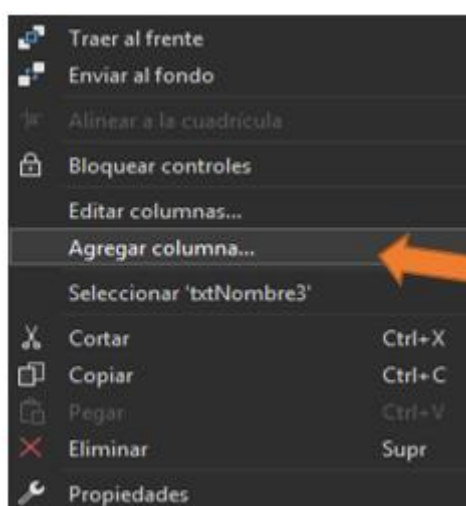
Figura 51. Conexión de Biométrico 2.



Como se observa los Biométricos tienen una dirección IP diferente, lo único que cambia es el cuarto Octeto y ya no hay coincidencias de dirección IP por lo tanto debemos tener datos satisfactorios en ambos Biométricos.

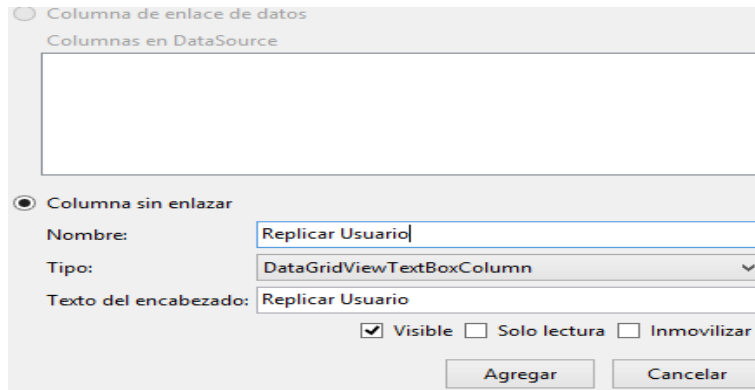
Para replicar un usuario en los Biométricos 1 y 2 en la tabla de Registrar se necesita colocar un Botón con la opción de agregar Columna como se muestra en la figura 52.

Figura 52. Señalización de opción para agregar Columna.



Cuando se obtiene la opción de agregar nueva columna despliega una Opción de Configuraciones como se muestra en la figura 53.

Figura 53. Configuraciones de la Nueva columna con Botón para Replicar.



En la tabla 14 se muestra el resultado después de agregar la columna.

Tabla 14. Columna Botón de nombre Replicar Usuario.

Numero de Registro	Nombre de trabajador	Contrasena	ID Privilegio	Replicar Usuario

Una vez que se obtiene el botón que tendrá la función para replicar usuario a Biométrico 2 se procede a la programación.

```
private void Click_Huelladigital(object sender, DataGridViewCellEventArgs e) {

    Console.WriteLine("*****");

}
```

Listo, una vez que se accede al código del Botón de la columna “Replicar Usuario” Arroja la función que se crea por defecto, pero asignando un nombre correspondiente a la función, en este caso la función se llama Click_Huelladigital como se muestra en el siguiente ejemplo:

```
private void Click_Huelladigital(object sender, DataGridViewCellEventArgs e)
```

Una vez que se tiene la función se agrega un evento de clic para un botón llamado “Run”, este evento (‘Run_Click’) a continuación se muestra un ejemplo de “Run” se activará cuando se haga un click asociado a la función que se requiere que funcione, pero se utiliza para ejecutar script de Python, entonces no se haya una forma de realizarlo con C#, por lo tanto, se necesita Instalar el lenguaje de Programación de Python, el conocimiento se obtuvo de la fuente (stackoverflow, 2022).

4.1 Instalación de Python y uso de clases de Python

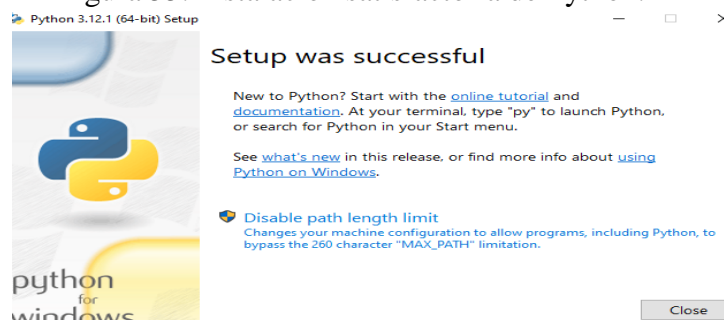
Para tener el funcionamiento del lenguaje de programación de Python se debe instalar y se comienza como se muestra en la figura 54.

Figure 54. .exe para instalación lenguaje Python.



Ahora se instala él .exe como se muestra en la figura 55.

Figura 55. Instalación satisfactoria de Python.



Una vez instalado el lenguaje de programación de Python, está listo para que se utilice, por el momento no es necesario utilizar un IDE en específico para programar funciones, uno de los IDEs más comunes es Visual Studio Code para Python o también Anaconda, pero se reitera, no es necesario, los códigos que se requieren invocar únicamente se modifican con editor de Texto debido a que los códigos son clases asignadas por el área de desarrollo TI de la empresa Centro de Validación Digital CVDSA, S.A. de C.V, la opción para instalar Python se obtuvo de la siguiente referencia (Python, 2022).

4,2 . Implementación del Evento Run_Click

Como ya se había mencionado el método Run_Click se utiliza para invocar códigos de Python, en este caso se necesita invocar el código de la opción de la Huella Digital al igual que el código de la voz del Biométrico MB160, se muestra un ejemplo de esta función en el siguiente ejemplo.

```
private void Run_Click(object sender, EventArgs e)

{

run_cmd("C:/Python34/python.exe", "C:/Users/Alaseel/Desktop/test.py");

}
```

La función está llamando a 'run_cmd' con dos argumentos: la ruta estable de Python.

La función 'run_cmd' con dos argumentos: La ruta ejecutable de Python (C:/Python34/python.exe) y la ruta de Python (C:/Users/Alaseel/Desktop/test.py), la función 'run_cmd' ejecuta un proceso de códigos en este caso el ejecutable de Python junto con el

Script como argumento, en este caso el Script que se necesita utilizar son las clases que se asignaron como ya se ha mencionado anteriormente, por el momento solo se ha mostrado un ejemplo de la función 'run_cmd' en la función Click_Huelladigital.

Ahora se implementa el evento de 'run_cmd' de la forma como se muestra en el siguiente ejemplo.

```
private void Click_Huelladigital(object sender, DataGridViewCellEventArgs e) {  
  
    Console.WriteLine("*****");  
    run_cmd("C:\\Users\\felipe\\Python\\Python312\\python.exe", "C:\\Users\\felipe\\Documents\\quadrum-biometrico-2022copia\\quadrum-biometrico-2022\\Huellas.py");  
  
    }  
  
}
```

Ahora lo que se observa es que se necesita ayuda de directorios, en este caso, se necesitan dos directorios, uno donde se encuentra el .exe del lenguaje de programación de Python y otro donde se encuentra el proyecto de la programación del Biométrico.

A continuación, se muestra el primer directorio que contiene el evento run_cmd:

```
("C:\\Users\\felipe\\Python\\Python312\\python.exe")
```

El directorio como ya se mencionó tiene la ubicación del .exe del lenguaje de programación de Python, para obtener una función del lenguaje de programación de Python, tal como se muestra en la figura 56.

Figura 56. Ubicación de Python.exe.

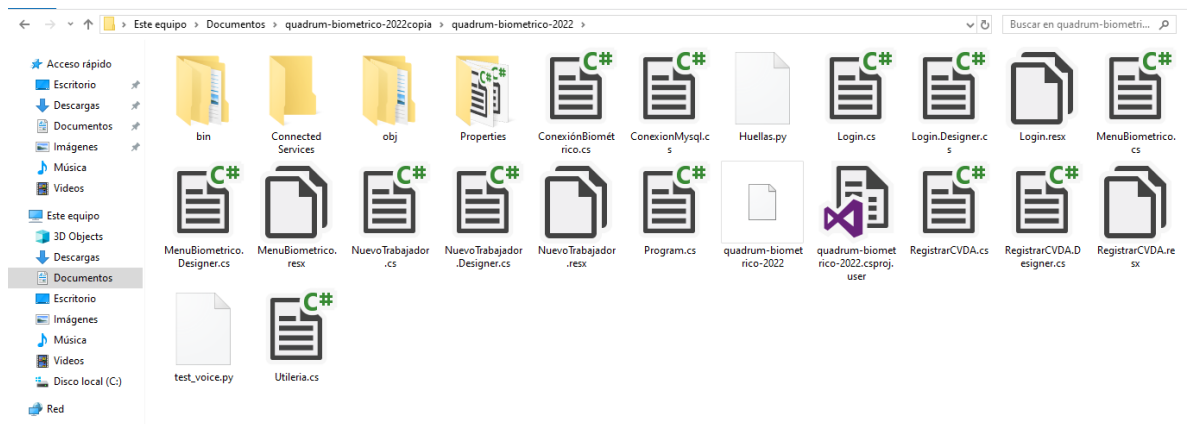
DLLs	01/04/2024 02:15 a. m.	Carpeta de archivos	
Doc	01/04/2024 02:15 a. m.	Carpeta de archivos	
include	01/04/2024 02:16 a. m.	Carpeta de archivos	
Lib	01/04/2024 02:17 a. m.	Carpeta de archivos	
libs	01/04/2024 02:17 a. m.	Carpeta de archivos	
Scripts	01/04/2024 02:17 a. m.	Carpeta de archivos	
tcl	01/04/2024 02:18 a. m.	Carpeta de archivos	
LICENSE	07/12/2023 10:21 p. m.	Documento de te...	37 KB
NEWS	07/12/2023 10:22 p. m.	Documento de te...	1,652 KB
python	07/12/2023 10:21 p. m.	Aplicación	101 KB
python3.dll	07/12/2023 10:21 p. m.	Extensión de la ap...	67 KB
python312.dll	07/12/2023 10:21 p. m.	Extensión de la ap...	6,840 KB
pythonw	07/12/2023 10:21 p. m.	Aplicación	100 KB
vcruntime140.dll	07/12/2023 10:21 p. m.	Extensión de la ap...	117 KB
vcruntime140_1.dll	07/12/2023 10:21 p. m.	Extensión de la ap...	49 KB

Ahora se muestra el segundo directorio que contiene el evento run_cmd:

"C:\\Users\\felipe\\Documents\\quadrum-biometrico-2022copia\\quadrum-biometrico-2022\\Huellas.py".

El directorio contiene la ubicación de todos los archivos que se han desarrollado en el proyecto incluyendo las clases de código del lenguaje de programación de Python que se utilizarán, en este caso solo se colocó una sola clase dentro del directorio, la clase de Huellas.py, con una clase que se encuentre dentro es más que suficiente para invocar las clases de Python, tal como se muestra en la figura 57.

Figura 57 Ubicación de archivos de Python.

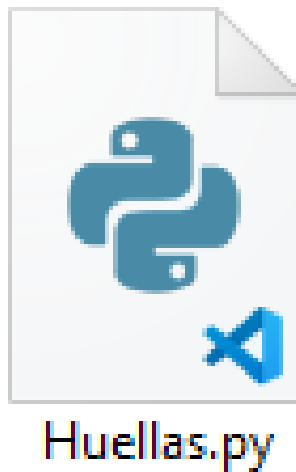


Como se puede observar ya se encuentran las clases de código de Python dentro de la carpeta del archivo las clases del proyecto, las clases son test_voice.py al igual que la clase de Huellas.py, listo ya se pueden invocar.

RESULTADOS Y DISCUSIÓN.

Para tener la réplica de la huella digital se utilizó modificar las clases de Python para obtener la réplica de los usuarios en un segundo Biométrico en este caso modificar las direcciones IP, en la figura 58 Se muestra la clase donde se modifican las direcciones IP para replicar el dato de huella digital de Python Huellas.py

Figura 58. Clase de Huellas.py.



Ahora se muestran las líneas de código de la clase Huellas.py junto con las explicaciones correspondientes.

```
import sys
```

```
import os
```

```
CWD = os.path.dirname(os.path.realpath(__file__))
```

```
ROOT_DIR = os.path.dirname(CWD)
```

```
sys.path.append(ROOT_DIR)
```

```
from zk import ZK, const
```

```

from zk.finger import Finger

from zk.user import User

conn = None

connDos = None

class Huellas(object):

    def __init__(self,zk):

        self.zk=zk

    def cambioBiometrico():

        conn = None

        zk = ZK('192.168.100.63', port=4370, timeout=5, password=0,
force_udp=False, ommit_ping=False)

        try:

            conn = zk.connect()

            conn.disable_device()

            users = conn.get_users()

            for user in users:

                print(user)

            conn94 = None

            zk94 = ZK('192.168.100.64', port=4370, timeout=5, password=0,
force_udp=False, ommit_ping=False)

```

```

conn94 = zk94.connect()

conn94.disable_device()

conn94.set_user(uid=user.uid, name=user.name,
privilege=user.privilege, password=user.password, group_id=user.group_id,
user_id=user.user_id, card=user.card)

template = conn.get_user_template(uid=user.uid, temp_id=6)

print(template)

conn94.save_user_template(user, template)

conn94.test_voice()

conn94.enable_device()

conn94.disconnect()

# Test Voice: Say Thank You

conn.test_voice()

# re-enable device after all commands already executed

conn.enable_device()

except Exception as e:

    print ("Process terminate: {}".format(e))

finally:

    if conn:

        conn.disconnect()

```

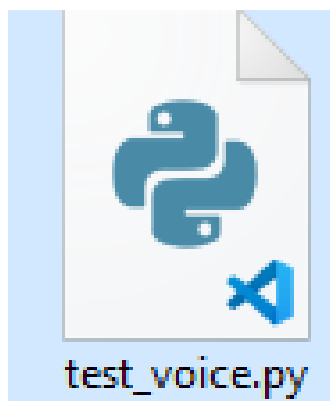
```
print(cambioBiometrico())
```

Se muestra la línea de código para el Biométrico 1 `zk = ZK('192.168.100.63', port=4370, timeout=5, password=0, force_udp=False, ommit_ping=False)` como se observa se agrega la dirección Ip del Biométrico1 '192.168.100.63', ya que es la dirección IP que se encuentra en la clase de Utilería en las funciones de los archivos de código del lenguaje C#.

Ahora se muestra la línea de código donde se encuentra la segunda dirección IP del Biométrico 2 `zk94 = ZK('192.168.100.64', port=4370, timeout=5, password=0,` como se observa la dirección Ip es 192.168.100.64, y también es la dirección IP que se encuentra dentro de la clase utilería de los archivos de las funciones de C#.

También se agrega la clase de voz en la clase de Python llamada `test_voice.py`, en la figura 59 se muestra la imagen de la clase de voz `tezt_voice`.

Figura 59 Imagen de clase de voz llamada `test_voice.py`



Ahora se muestran las líneas de código de la clase `test_voice.py`

```
# -*- coding: utf-8 -*-
```

```
import os

import sys

import time

CWD = os.path.dirname(os.path.realpath(__file__))

ROOT_DIR = os.path.dirname(CWD)

sys.path.append(ROOT_DIR)

from zk import ZK

zk = ZK('192.168.100.63', port=4370)

try:

    conn = zk.connect()

    for i in range(0, 55):

        print ("Voice number #%d" % i)

        conn.test_voice(i)

        time.sleep(3)

except Exception as e:

    print ("Process terminate : {}".format(e))

finally:

    if conn:

        conn.disconnect()
```


La línea de código que contiene la dirección IP es la siguiente:

```
zk = ZK('192.168.100.63', port=4370)
```

Como se observa la dirección Ip es 192.168.100.63, en esta clase solo es necesario colocar una Ip, ya que en la clase anterior de las Huellas.py ya se encuentran las dos Ip que se necesitan para el Biométrico 1 y el Biométrico 2, esta IP se encuentra en la clase de las huellas al igual que en la clase Utileria ubicada en los archivos del código de C#, en el siguiente ejemplo se muestran las funciones realizadas en C#.

```
public static zkemkeeper.CZKEM coneccionBiometrico1()
{
    zkemkeeper.CZKEM axCZKEM1 = new zkemkeeper.CZKEM();
    axCZKEM1.Connect_Net("192.168.100.63", 4370);
    return axCZKEM1;
}

//Metodo para Biometrico 2

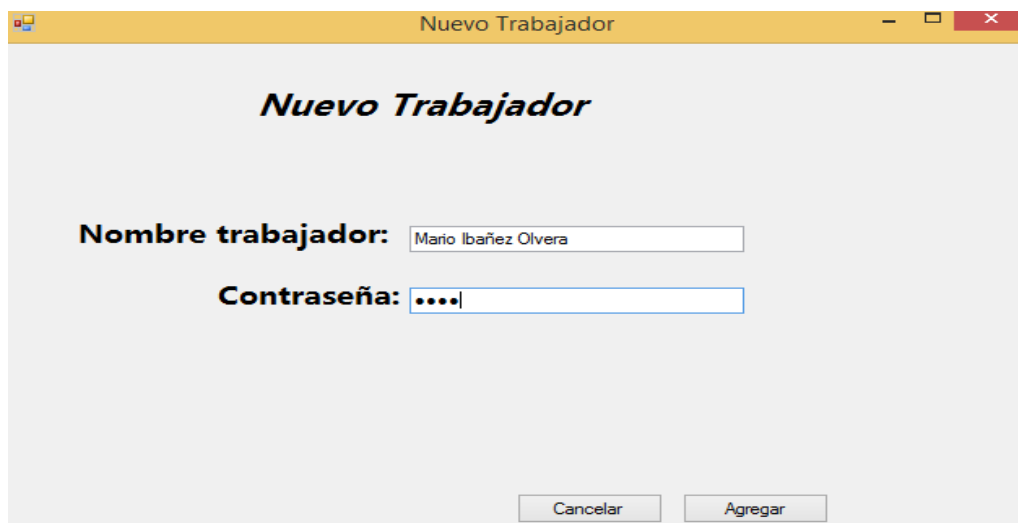
public static zkemkeeper.CZKEM coneccionBiometrico2()
{
    zkemkeeper.CZKEM axCZKEM2 = new zkemkeeper.CZKEM();
    axCZKEM2.Connect_Net("192.168.100.64", 4370);
    return axCZKEM2;
}
```

}

Como se puede observar las direcciones Ip coinciden con las direcciones IP de las clases de Python de nombre Huellas.py, así como la clase test_voice.py.

Ahora para comprobar que funciona lo que ya se explicó, se realiza un registro de un usuario se accede a la ventana de nuevo trabajador agregando el nombre Mario Ibáñez Olvera como se muestra en la figura 60.

Figura 60. Usuario agregando nuevo usuario

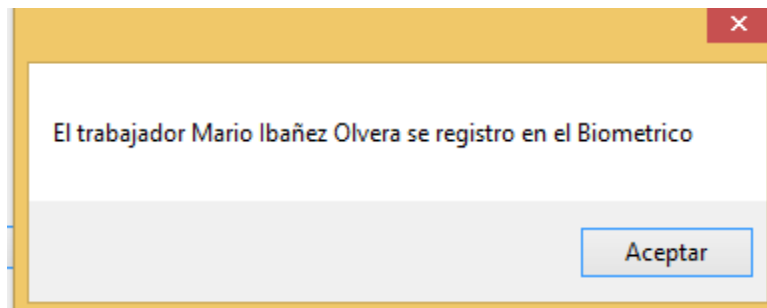


The image shows a screenshot of a web application window titled "Nuevo Trabajador". The window has a yellow header bar with the title and standard window controls (minimize, maximize, close). The main content area is light gray and contains the following elements:

- The title "Nuevo Trabajador" in a bold, italicized black font.
- A label "Nombre trabajador:" followed by a text input field containing the text "Mario Ibáñez Olvera".
- A label "Contraseña:" followed by a password input field containing four dots "....".
- At the bottom of the form, there are two buttons: "Cancelar" and "Agregar".

Ahora se presiona el botón de agregar y se obtiene un registro satisfactorio como se muestra en la figura 61.

Figura 61. Mario Ibáñez Olvera registrado satisfactoriamente.



Una vez que se arrojan datos satisfactorios de Registro se procede a la consulta donde está registrado el usuario nuevo, en la tabla 15 Se muestra el nuevo usuario registrado.

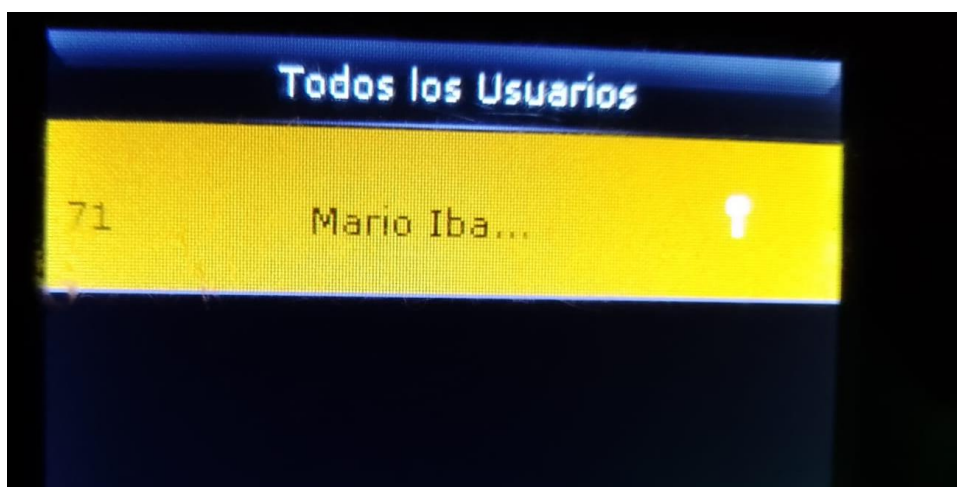
Tabla 15. Nuevo usuario registrado

	Numero de Registro	Nombre de trabajador	Contraseña	ID Privilegio	Replicar Usuario
▶	71	Mario Ibañez Olvera	1234	privilegio	Replicar Huella
*					

Numero de Registro	Nombre de trabajador	Contraseña	ID Privilegio	Replicar Usuario
71	Mario Ibáñez Olvera	1234	Privilegio	Replicar Huella

Ahora se consulta en el Biométrico 1 que el usuario se encuentre registrado, en la figura 62 se muestran los resultados del usuario registrado.

Figura 62. Usuario registrado en el Biométrico

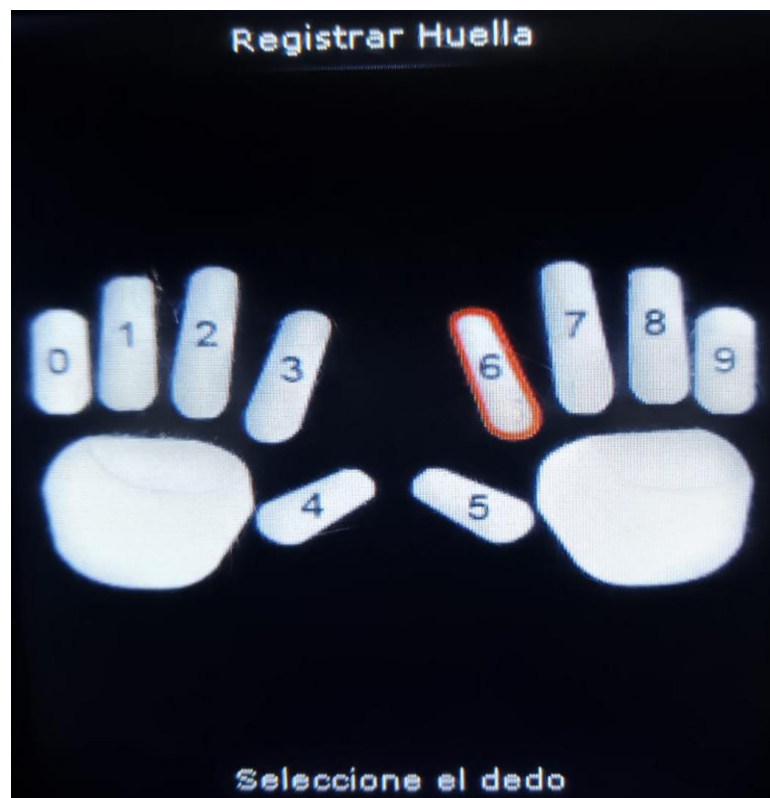


La figura anterior únicamente muestra al lado del Nombre una imagen pequeña de llave, la cual esa imagen pequeña, únicamente representa que el usuario se encuentra con su dato de contraseña, por el momento no contiene Huella digital.

Para agregar una huella digital no fue posible realizar una función a través del código, de C# ni mucho menos con el código de Python, únicamente se puede agregar de manera manual, se puede colocar cualquier la huella del dedo de las manos izquierda o derecha, ya es de acuerdo con el gusto de cada usuario, cualquier huella de dedo se puede guardar y es replicable.

Las huellas de los dedos se miden del 0 al 9 como se muestra en la figura 63 por lo tanto, es criterio de cada usuario cual huella desea guardar en el biométrico para después replicarlo.

Figura 63. Huellas digitales de Biométrico.



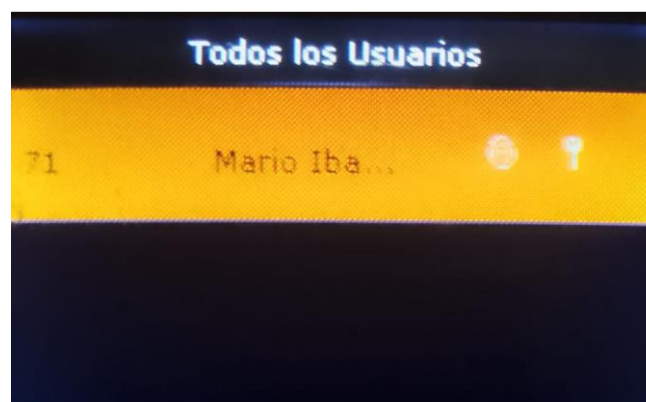
En la figura 63 se muestra cómo se coloca el dedo en el lector del Biométrico, se eligió el dedo índice de la mano derecha con número 6, tal como se muestra en la siguiente figura 64.

Figura 64. Agregando huella digital en Biométrico manera Manual.



Una vez que se agrega la huella digital se guarda en el usuario registrado aparece de la forma tal y como se muestra en la figura 65.

Figura 65, Imagen de usuario con huella y contraseña.



Como se observa en la figura anterior al lado del nombre del usuario, se encuentra ahora con dos dibujos, el primero una huella digital y el segundo con una llave que representa la contraseña de dato de usuario, todo en el Biométrico 1.

Ahora se visualiza si también se encuentra en la base de datos el usuario que se registró, en la tabla 16 Se observa la base de datos.

Tabla 16. Base de datos con usuario registrado en la base de datos.

id	nombre	contrasena	Huella_digital
71	Mario Ibáñez Olvera	1234	

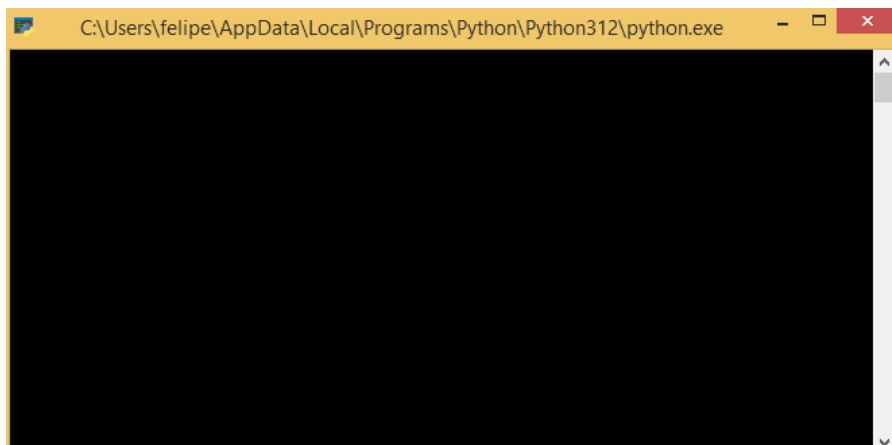
Una vez que se tiene registrado el usuario de la persona que se registró, se presiona el botón de replicar Huella, como se muestra en la tabla 17.

Tabla 17. se muestra que se ha presionado el botón de replicar Huella.

Numero de Registro	Numero de trabajador	Contraseña	ID Privilegio	Replicar Usuario
71	Mario Ibáñez Olvera	1234	privilegio	Replicar Huella

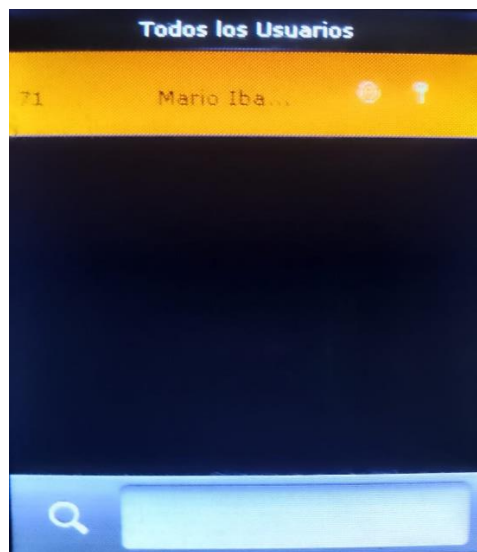
Una vez que se presiona el botón de replicar huella se despliega una ventana de consola de Python como se muestra en la figura 66.

Figura 66. Consola de lenguaje de programación de Python.



Satisfactoriamente se obtuvo lo que se necesitaba, en la figura 67 se muestra el resultado de lo que se necesitaba.

Figura 67. Datos satisfactorios de la réplica del Biométrico 2.



Conclusiones y recomendaciones.

Con el lenguaje de programación C#, el gestor de base de datos de MySQL se logró crear la base de datos con el registro del usuario, contraseña y huella digital funcionando correctamente, desde el lenguaje de programación C#, todo esto incluyendo que el ID de cada trabajador que se registra de manera automática como nuevo en la empresa Centro de Validación Digital CVDSA, S.A. de C.V, con las pruebas que se realizaron se obtuvieron resultados, el registro de nuevos trabajadores de la empresa, en biométrico y base de datos.

Para el método de huella digital, se logró desarrollar un método que invoca líneas de código del lenguaje de programación Python. Se utilizó un método que ayudó a guardar la huella digital, únicamente en los Biométricos, ya que fue lo más complejo del proyecto.

Si fue posible crear el conector para interactuar con el Hardware Biométrico, todo esto por medio de la dirección IP del Router de internet vinculado con el Biométrico controlado desde el lenguaje de programación de C#, porque debido a que el lenguaje de programación de C# es un lenguaje de programación tipo lenguaje ensamblador de alto nivel. este Por medio del lenguaje de programación nos sirvió mucho porque se crean métodos para configurar la conexión del Biométrico con la Ip, fue totalmente posible realizar el conector, prácticamente el conector se realizó con un código de programación llamado utilería, ahí se crearon los métodos para crear la conexión con el Biométrico.

Si se llevó a cabo el desarrollo de la consulta del registro de los usuarios junto con la base de datos de MySQL, una vez creado se puede modificar y eliminar dentro de la base de datos, y visualizado en la consulta, en el biométrico no fue posible visualizar la eliminación o modificación de algún usuario, porque en el momento de correr el programa afectaba para el funcionamiento debido a que se tienen dos procesos vinculados la base de datos y los Biométricos.

Si se logró extraer los datos de los trabajadores desde el biométrico, con id, nombre, contraseña huella digital, en el momento que se accede con la huella digital del trabajador,

inmediatamente el biométrico manda el mensaje de voz de acceso correcto, al igual que con la réplica del Biométrico número 2.

REFERENCIAS

{lwp} comunidad de programadores. (8 de Julio de 2022). Obtenido de {lwp} comunidad de programadores: <https://www.lawebdelprogramador.com/foros/Visual-Basic.NET/902590-Biometria-Huella-Digital.html>

6.3, M. W. (s.f.). *UDGVIRTUAL*. Obtenido de UDGVIRTUAL.

ADVANCE, R. C. (2 de abril de 2022). *RJ CODE ADVANCE*. Obtenido de <https://rjcodeadvance.com/cap-2-login-logout-y-mostrar-datos-del-usuario-validaciones-con-arquitectura-en-capas-poo-c-y-mysql-nivel-intermedio/>

ALEGSA.com.ar. (17 de 06 de 2022). Obtenido de ALEGSA.com.ar: <https://www.alegsa.com.ar/Dic/login.php#gsc.tab=0>

anixter, W. (12 de Marzo de 2022). *WESCO anixter*. Obtenido de Biometría vascular ¿Es el futuro?: https://www.anixter.com/es_mx/about-us/news-and-events/news/vascular-biometrics-is-it-the-future.html

CEBALLOS SIERRA, F. (2011). *MICROSOFT C#, PROGRAMACIÓN ORIENTADA A OBJETOS*. MEXICO: RA-MA.

Cero, P. D. (2 de abril de 2022). *Conexión a Base de Datos MySQL con C#*. Obtenido de <https://programacion.top/csharp-mysql/>

CORAL, L. Á. (12 de Abril de 2022). *EL TIEMPO*. Obtenido de <https://www.eltiempo.com/tecnosfera/apps/visual-studio-code-es-un-editor-de-codigo-fuente-de-microsoft-761049#:~:text=Visual%20Studio%20Code%20es%20un%20software%20de%20Microsoft%20que%20edita,compatible%20con%20JavaScript%20y%20Node.>

Cosio, N. A. (2013). *GGUIA TOTAL DEL PROGRAMADOR*. BUENOS AIRES ARGENTINA: ReduSERS.

DOUGLAS BELL, M. P. (2010). *C# PARA ESTUDIANTES*. ESCUELA INGENIERIA UNIVERSIDAD PANAMERICANA , MEXICO: PEARSON.

Ejemplos de ZK en Python. (23 de JUNIO de 2022). Obtenido de Ejemplos de ZK en Python: <https://python.hotexamples.com/es/examples/zk3/ZK/-/python-zk-class-examples.html>

Ejemplos de ZK.VERSION en Java. (3 de junio de 2022). Obtenido de Ejemplos de ZK.VERSION en Java: <https://java.hotexamples.com/es/examples/-/ZK.VERSION/-/java-zk.version-class-examples.html>

freelancermap. (13 de enero de 2022). Obtenido de freelancermap: <https://www.freelancermap.com/blog/es/que-hace-programador-delphi/>

GILFILLAN, L. (2020). LA BIBLIA MySQL. En L. GILFILLAN, *LAN GILFILLAN* (pág. 880). ANAYA MULTIMEDIA.

Jeff Ferguson, B. P. (2003). *LA BIBLIA DE C#*. ESPAÑA: ANAYA.

kaspersky VPN Secure Connection. (s.f.). Obtenido de <https://latam.kaspersky.com/resource-center/definitions/what-is-an-ip-address>

LUIS ALBERTO CASILLAS SANTILLÁN, M. A. (2012). Bases de datos en MySQL. MEXICO: UOC FORMACIÓN DE POSGRADO.

LUIS ALBERTO CASILLAS SANTILLÁN, M. G. (2010). Bases de datos en MySQL,. México: UOC Formación de Posgrado.

MB160, Z. (s.f.). *zk-software.com*. Obtenido de zk-software.com: <https://zk-software.com/reloj-chedador-rostro-zk-mb160.html>

MDNWEBDOCS. (s.f.). Obtenido de MDNWEBDOCS: <https://developer.mozilla.org/es/docs/Glossary/CRUD>

Microsoft. (23 de Marzo de 2022). *Novedades de Visual Studio 2022*. Obtenido de Novedades de Visual Studio 2022: <https://learn.microsoft.com/es-es/visualstudio/ide/whats-new-visual-studio-2022?view=vs-2022>

Modernización de Sistemas Legados. (10 de Agosto de 2022). Obtenido de Modernización de Sistemas Legados: <https://www.go4it.solutions/es/blog/las-herramientas-que-necesitas-para-programar-en-powerbuilder>

MuaazH. (10 de junio de 2022). Obtenido de MuaazH: https://github.com/MuaazH/ZKTeco_PULLSDK_Wrapper

Muy Tecnológicos. (s.f.). Obtenido de <https://muytecnologicos.com/diccionario-tecnologico/ip-publica>

Ortiz, S. A. (7 de JUNIO de 2022). *QUORA*. Obtenido de QUORA: <https://es.quora.com/En-qu%C3%A9-lenguaje-de-programaci%C3%B3n-se-hacen-los-detectores-biom%C3%A9tricos>

Python. (11 de Junio de 2022). *Python*. Obtenido de Python: <https://www.python.org/downloads/>

pyzk. (10 de Junio de 2022). Obtenido de pyzk: <https://github.com/fananimi/pyzk>

SANTANA, M. A. (2018). TESIS RECONOCIMIENTO POR BIOMETRÍA FACIAL PARA. En R. P. PARA, *TESIS RECONOCIMIENTO POR BIOMETRÍA FACIAL PARA* (pág. 56). El Cerrillo Piedras Blancas, Toluca, Estado de México.: TESIS.

Sistema de captura de muestras. (2017). En M. P. Montes, *Sistema de captura de muestras* (pág. 77). Madrid España.

stackoverflow. (12 de Mayo de 2022). *stackoverflow*. Obtenido de stackoverflow: <https://stackoverflow.com/questions/36136861/trying-to-call-python-script-in-c-sharp>

stackoverflow. (2 de Julio de 2022). *stackoverflow*. Obtenido de stackoverflow: <https://es.stackoverflow.com/questions/tagged/zkemkeeper>

SYSCOM. (12 de JUNIO de 2022). *Enrolador de huellas USB de alta resolución / SDK gratuito para desarrollos propios (JAVA, ANDROID, Windows C#) / Compatible con software ZKTeco*. Obtenido de Enrolador de huellas USB de alta resolución / SDK gratuito para desarrollos propios (JAVA, ANDROID, Windows C#) / Compatible con software ZKTeco: <https://www.syscom.mx/producto/ZK-9500-ZKTECO-160890.html>

Teaming. (10 de enero de 2022). Obtenido de Teaming: <https://www.clubdelphi.com/foros/showthread.php?t=87460>

ZKT. (26 de 06 de 2022). *ZKT (ECO) PREMIUM*. Obtenido de ZKT (ECO): <https://desarrollo.zktecolatinoamerica.com/downloads>