



TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE APIZACO

DEPARTAMENTO DE SISTEMAS Y COMPUTACIÓN
INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN Y
COMUNICACIONES

PROYECTO DE RESIDENCIAS
PROFESIONALES

NOMBRE DEL PROYECTO:

Análisis y Experimentación para Detección y Reconocimiento de
Huellas Dactilares

PRESENTA

ARMANDO ARTURO ZENTENO LIMA

ASESOR EXTERNO

Dr. Alan Augusto Gallegos Cuéllar

ASESOR INTERNO

M.I.A. Carlos Pérez Corona



índice

1. Agradecimientos	5
2. Resumen	6
3. Introducción.....	7
4. Generalidades de la empresa	8
4.1 Mapa de la empresa	8
4.2 Logo de la empresa	9
4.3 Organigrama de la empresa	9
4.4 Acerca del Tecnológico Nacional de México / Instituto Tecnológico de Apizaco	10
4.5 Misión y Visión	10
4.6 VALORES	10
5. Delimitación del problema	11
6. Objetivos	12
6.1 Objetivo General	12
6.2 Objetivos Específicos:	12
7. Justificación	13
8. Organigrama	14
8.1 Descripción Detallada de las Actividades	15
9. Marco Teórico.....	16
9.1 Huellas dactilares	16
9.1.1 Identificación de Huellas Dactilares	16
9.1.2 Funciones de Huellas Dactilares	17
9.1.3 Patrones de Huellas Dactilares	18
9.1.4 Huellas Dactilares Distintivas y Persistentes	19
9.2 ingeniería Inversa	20
9.2.1 Decompiladores	21
9.2.2 Archivos .DLL.....	21
10. Herramientas Usadas	23
10.1 Herramientas Físicas (hardware)	23



10.1.1 Lector Biométrico U.are.U 4500 Reader	23
10.2 Herramientas Digitales (software).....	24
10.2.1 Visual Basic	24
10.2.2 C#	26
10.2.3 Bibliotecas DLL	27
10.2.4 HTML.....	28
10.2.5 Bootstrap.....	30
10.2.4 Python.....	31
10.2.5 Python.NET.....	32
10.2.6 Visual Studio.....	33
10.2.7 Visual Studio Code	34
10.2.8 DotPeek.....	35
10.2.9 SDK Digital Persona	36
10.2.10 Django	37
11. Desarrollo del Proyecto	40
11.1 Instalaciones	40
11.1.1 Digital Persona SDK	41
11.1.2 Visual Studio.....	42
11.1.3 DotPeek.....	43
11.1.4 Visual Studio Code	44
11.1.5 Python.....	45
11.1.6 Pythonnet.....	46
11.2. Desarrollo del Proyecto	47
11.2.1 Biblioteca Dll	47
11.2.2 Entorno Virtual.....	50
11.2.3 Programación en Django	51
11.2.3 Imágenes.....	58
11.2.4 Plantillas	59
12. Resultados Obtenidos.....	62



13. Conclusiones	69
14. Competencias Desarrolladas	70
15. Referencias	71



1. Agradecimientos

Agradezco sinceramente a Dios por darme la oportunidad de haber concluido con éxito esta etapa de mi vida, agradezco a mi familia y amigos que, con su ayuda invaluable, contribuyeron a que esto fuera posible.

Quiero agradecer especialmente a, Heather, mi pequeña hija, quien fungió como el motor imparable, no solo de esta meta, sino de mi vida entera. Me has acompañado desde tercer semestre hasta el día de hoy. Primero dios, yo igual estaré a tu lado apoyándote en todo lo que pueda cuando estes dentro de tus estudios profesionales.

También deseo expresar mi gratitud a la directora del Instituto Tecnológico de Apizaco, Yesica Imelda Saavedra Benítez, por su invaluable liderazgo, al jefe del departamento de Sistemas y Computación, Alan Augusto Gallegos Cuéllar, quien ha desempeñado un papel crucial como asesor externo en este proyecto. Sus conocimientos y orientación han enriquecido significativamente mi trabajo.

Asimismo, agradezco al profesor Carlos Pérez Corona, mi asesor interno, por su guía experta y su compromiso con mi crecimiento académico. Su apoyo ha sido fundamental para alcanzar los objetivos de este proyecto.

Finalmente, me gustaría destacar el sentido de colaboración y camaradería que caracteriza al Instituto Tecnológico de Apizaco. La atmósfera de respeto y apoyo mutuo ha sido esencial en mi crecimiento tanto personal como profesional. Agradezco sinceramente por haber tenido la oportunidad de ser parte de esta comunidad universitaria tan acogedora y enriquecedora.

2. Resumen

El proyecto se basó en la creación de un sistema para analizar y experimentar con la detección y reconocimiento de huellas dactilares. Esto se logró mediante el uso principalmente de Python y Visual Basic como lenguajes de programación. La combinación de estos dos entornos permitió sacar provecho de sus respectivas fortalezas y características únicas para obtener un rendimiento óptimo. El objetivo principal era mejorar los procedimientos de identificación biométrica mediante la optimización de las características extraídas de las huellas dactilares.

Durante el proyecto, se llevó a cabo una investigación exhaustiva para evaluar diversos métodos de detección y reconocimiento con el fin de aumentar la precisión y eficacia en el reconocimiento de huellas dactilares. Se utilizó un enfoque meticuloso para analizar los resultados mediante métricas específicas, y se realizaron ajustes continuos en el código para mejorar el rendimiento del sistema.

Para la implementación de este proyecto, se utilizaron varias herramientas y software, incluyendo Visual Studio, Visual Studio Code y Python. Estas plataformas proporcionaron un entorno de desarrollo sólido y flexible para la implementación de algoritmos relacionados con la extracción y selección de características. Se aprovecharon las capacidades avanzadas de desarrollo y la integración de interfaces gráficas proporcionadas por estos entornos. Python se utilizó específicamente para programar algoritmos relacionados con la detección y reconocimiento de huellas dactilares, aprovechando su flexibilidad y poder en el procesamiento de imágenes y algoritmos de reconocimiento.

Además, se utilizaron librerías DLL (Dynamic Link Libraries) para acceder a funciones específicas y optimizar el rendimiento del sistema. Estas librerías permitieron integrar funcionalidades adicionales y mejorar la interoperabilidad entre los diferentes componentes del proyecto.

El proceso de experimentación se llevó a cabo mediante la ejecución de algoritmos diseñados para extraer y seleccionar características relevantes de las huellas dactilares. Se realizaron pruebas exhaustivas utilizando conjuntos de datos variados para evaluar el rendimiento y la precisión del sistema en diferentes escenarios.



3. Introducción

La utilización de la identificación biométrica, especialmente el reconocimiento de huellas dactilares ha surgido como una herramienta esencial en diversos ámbitos, desde la seguridad hasta la gestión de identidades. La singularidad y estabilidad de las huellas dactilares las convierten en un medio confiable para autenticar individuos. No obstante, para lograr un reconocimiento preciso y eficaz, es crucial realizar un análisis detallado y una selección cuidadosa de características.

El objetivo principal de este proyecto es mejorar la precisión y confiabilidad de los sistemas de identificación biométrica basados en huellas dactilares a través del análisis y experimentación para su detección y reconocimiento. Mediante la combinación de técnicas avanzadas de procesamiento de imágenes y algoritmos específicos, buscamos desarrollar una solución que supere los desafíos actuales en este campo, como la variabilidad de las huellas y el ruido en las imágenes capturadas.

En este marco, exploraremos diversos enfoques y metodologías para extraer y seleccionar características, evaluando su eficacia mediante experimentos rigurosos y análisis comparativos. Además, aprovecharemos herramientas y tecnologías de última generación, como Python y Visual Studio, para implementar y optimizar nuestro sistema.

El resultado de este proyecto no solo impulsará el avance en la tecnología de reconocimiento de huellas dactilares, sino que también tendrá aplicaciones prácticas en múltiples áreas, desde la seguridad en aeropuertos y fronteras hasta la gestión de acceso en entornos gubernamentales y corporativos. En resumen, nuestro objetivo es crear una solución sólida y confiable que mejore de manera significativa la precisión y eficiencia de los sistemas biométricos basados en huellas dactilares.

4. Generalidades de la empresa

Empresa: **Tecnológico Nacional de México / Instituto Tecnológico de Apizaco**

Dirección: Av Instituto Tecnológico s/n Conurbado Tzompantepec, Cdad. de Apizaco, Tlax.

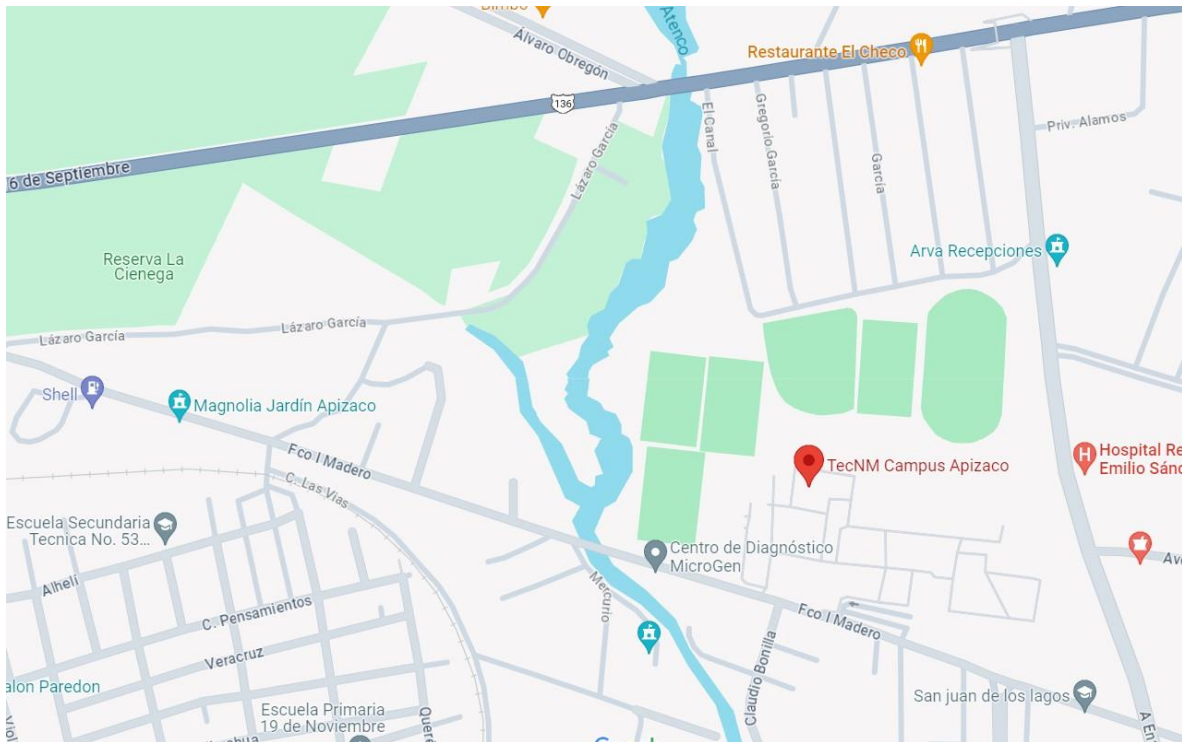
Código Postal: 90300

Teléfono: 241 417 2010

Correo: direccion@apizaco.tecnm.mx

Web: <https://www.apizaco.tecnm.mx/>

4.1 Mapa de la empresa

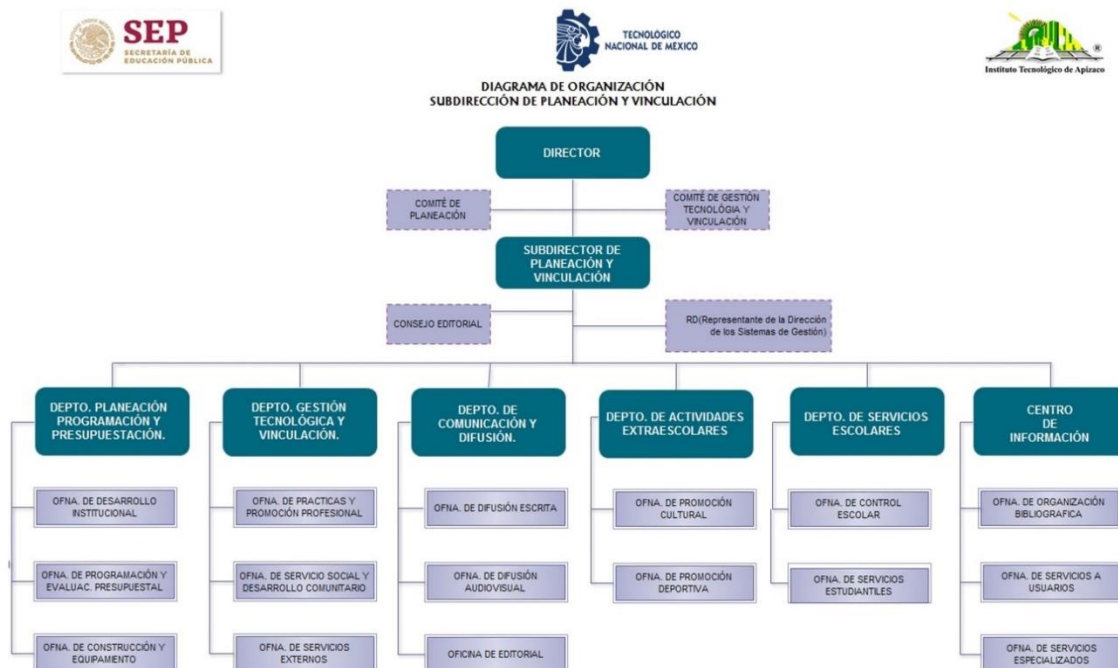


4.2 Logo de la empresa



4.3

Organigrama de la empresa



4.4 Acerca del Tecnológico Nacional de México / Instituto Tecnológico de Apizaco

El Tecnológico Nacional de México/Instituto Tecnológico de Apizaco, establece el compromiso de implementar y orientar todos sus procesos estratégicos y actividades hacia la calidad del servicio educativo, respeto del medio ambiente, seguridad y salud de las personas dando cumplimiento a los requisitos del estudiantado y partes interesadas, legislación aplicable y otros requisitos que se describan, así como promover en su personal, estudiantes y partes interesadas la prevención de la contaminación y el uso racional de los recursos, así como la prevención de accidentes, mediante la implementación, operación y mejora continua de un Sistema de Gestión Integral, conforme a las Normas ISO 9001:2015/NMX-CC9001-IMNC-2015, ISO 14001:2015/NMX-SAAIMNC-14001-2015 e ISO 45001:2018/NMX-SAST-45001/IMNC-2018, coadyuvando a la conformación de una sociedad justa y humana, con una perspectiva de sustentabilidad y de seguridad en el trabajo, siendo uno de los pilares fundamentales del desarrollo sostenido y sustentable.

4.5 Misión y Visión

MISIÓN: Formar profesionistas que sean líderes visionarios, emprendedores, con capacidad de generar soluciones disruptivas, innovando en la ciencia y la tecnología, comprometidos con el desarrollo económico, responsabilidad social teniendo una visión sostenible y sustentable.

VISIÓN: Ser una Institución de educación superior tecnológica de vanguardia, formadora de ciudadanos del mundo, con conocimientos intercultural, respetando la diversidad social.

4.6 VALORES

HONESTIDAD: Se demuestra día a día a través de la transparencia con la ejecución de las actividades laborales cotidianas.

RESPONSABILIDAD: Es la actitud que caracteriza al personal en el cumplimiento de su tarea con eficacia y eficiencia en beneficio de la comunidad.

LEALTAD: Es la actitud que caracteriza al personal del Instituto Tecnológico de Apizaco, aplica sus conocimientos y habilidades en el logro de las metas institucionales, sin olvidar en ningún momento el lema

5. Delimitación del problema

El presente proyecto tiene como objetivo analizar la información proporcionada por un sensor para procesarla y poder mejorar la implementación de la detección de huellas.

El proyecto se enfocará en la implementación de un sistema de detección y reconocimiento de huellas dactilares utilizando un sensor biométrico específico.

Se abordará la identificación de huellas dactilares como una forma de autenticación biométrica para acceder a sistemas o dispositivos.

La investigación se centrará en el desarrollo de algoritmos y técnicas de procesamiento de imágenes para analizar las huellas dactilares capturadas por el sensor y realizar su reconocimiento de manera eficiente y precisa.

Se considerará la integración del sistema con software complementario para gestionar la información de las huellas dactilares y realizar las correspondientes validaciones.

El proyecto se concentrará en el reconocimiento de huellas dactilares en condiciones normales de uso, excluyendo situaciones extremas como huellas parcialmente ilegibles o dañadas.

Se limitará la investigación al uso de un sensor de huellas digitales específico, sin explorar la compatibilidad con otros tipos de sensores biométricos.

La implementación del sistema estará restringida a entornos controlados para garantizar la precisión de los resultados, sin considerar desafíos adicionales asociados con ambientes no controlados o variables externas como iluminación o calidad de la piel.

La evaluación del sistema se realizará en base a métricas de precisión y velocidad, sin abordar aspectos relacionados con la seguridad de los datos biométricos almacenados o la interoperabilidad con sistemas de seguridad existentes.

Se realizarán pruebas exhaustivas para evaluar la precisión y robustez del sistema de reconocimiento de huellas dactilares, utilizando conjuntos de datos de prueba representativos y escenarios simulados de uso.

Se validarán los resultados obtenidos mediante comparaciones con sistemas de referencia o estándares reconocidos en el campo de la biometría, asegurando la fiabilidad y validez de los resultados obtenidos.

6. Objetivos

6.1 Objetivo General

Desarrollar un sistema de detección y reconocimiento de huellas dactilares que sea preciso, eficiente y confiable. Se busca poder contribuir al avance en el campo de la biometría y la seguridad informática, proporcionando una solución innovadora y efectiva para la detección y reconocimiento de huellas dactilares que pueda tener aplicaciones prácticas en diversos ámbitos, como la seguridad, la identificación personal y el control de acceso.

6.2 Objetivos Específicos:

- Investigar y analizar las técnicas y algoritmos más avanzados en el campo de la biometría y el procesamiento de imágenes aplicados a la detección y reconocimiento de huellas dactilares.
- Diseñar e implementar un sistema que permita la captura, almacenamiento y análisis de huellas dactilares utilizando un sensor biométrico adecuado.
- Desarrollar algoritmos de preprocesamiento de imágenes para mejorar la calidad de las huellas dactilares y eliminar posibles ruidos o distorsiones.
- Implementar algoritmos de reconocimiento biométrico que permitan identificar y autenticar a individuos basándose en las características únicas de sus huellas dactilares.
- Validar y evaluar el sistema mediante pruebas exhaustivas para medir su precisión, velocidad y robustez en diferentes condiciones de uso y con diferentes conjuntos de datos.
- Proponer mejoras y ajustes al sistema basados en los resultados obtenidos de las pruebas, con el objetivo de optimizar su desempeño y fiabilidad.
- Documentar todo el proceso de investigación, desarrollo e implementación del sistema en un informe de tesis que cumpla con los estándares académicos y científicos.



7. Justificación

En la actualidad, la biometría se ha convertido en un área crucial de investigación y desarrollo en el campo de la seguridad informática y la autenticación de identidades. El uso de huellas dactilares como método de identificación biométrica es ampliamente aceptado debido a su singularidad, estabilidad y facilidad de adquisición. Un proyecto centrado en la detección y reconocimiento de huellas dactilares contribuiría directamente al avance y la mejora de sistemas de seguridad basados en esta tecnología, como el acceso seguro a dispositivos, la identificación personal en sistemas gubernamentales y la gestión de la seguridad en diversas industrias. La investigación en el campo de la biometría y el procesamiento de imágenes ofrece oportunidades significativas para desarrollar y aplicar algoritmos avanzados, técnicas de aprendizaje automático y análisis de datos. Un proyecto de tesis en detección y reconocimiento de huellas dactilares permitiría explorar y contribuir a la evolución de estas áreas, generando conocimientos valiosos y metodologías innovadoras que puedan ser aplicadas en otros contextos biométricos y de seguridad.

[illegible]



8.1 Descripción Detallada de las Actividades

Nombre de la Tarea	Duración	Comienzo	Fin
Inicio	10 días	1/09/2023	11/09/2023
Introducción	10 días	11/09/2023	21/09/2023
Asignación de del Proyecto	15 días	22/09/2023	07/10/2023
Definición de Requisitos y alcance	8 días	08/10/2023	16/10/2023
Planificación del proyecto	20 días	17/10/2023	04/11/2023
Diseño del sistema	8 días	05/11/2023	13/11/2023
Desarrollo del software	15 días	14/11/2023	29/11/2023
Entrenamiento del modelo de Reconocimiento	25 días	30/11/2023	24/12/2023

9. Marco Teórico

9.1 Huellas dactilares

Una huella dactilar es una marca única y distintiva que se encuentra en la superficie de la piel de los dedos de las manos, específicamente en las yemas de los dedos. Estas marcas se forman debido a la disposición de crestas y surcos en la epidermis de la piel, creando patrones que son únicos para cada individuo, incluso entre gemelos idénticos.

Las huellas dactilares son consideradas como una característica biométrica confiable y única en cada persona. La singularidad de las huellas dactilares radica en la combinación única de características como bifurcaciones, terminaciones, puntos característicos y otros elementos, que forman patrones complejos y específicos en cada dedo. Estos patrones son estables a lo largo de la vida de una persona y no cambian, lo que las hace extremadamente valiosas para la identificación y autenticación de individuos en diversas aplicaciones, como la seguridad, la identificación personal y la investigación forense.

El análisis y la comparación de las huellas dactilares se llevan a cabo mediante tecnologías biométricas que permiten registrar y almacenar las características únicas de las huellas para su posterior uso en la verificación de identidad. La captura de huellas dactilares se realiza a través de sensores biométricos ópticos, capacitivos o ultrasonidos, que registran las crestas y surcos de las huellas para su análisis y comparación con una base de datos de huellas dactilares previamente registradas.

9.1.1 Identificación de Huellas Dactilares

La identificación de huellas dactilares se basa en varios fundamentos clave que hacen que este método sea confiable y efectivo para la autenticación biométrica. Aquí están algunos de esos fundamentos:

- **Unicidad:** Cada persona tiene un conjunto único de huellas dactilares. Incluso entre gemelos idénticos, las huellas dactilares son diferentes. Esto se debe a la combinación única de patrones de crestas, surcos, bifurcaciones, terminaciones y otros detalles que se forman en la piel de los dedos durante el desarrollo fetal y se mantienen constantes a lo largo de la vida de la persona.

- **Permanencia:** Las características de las huellas dactilares son estables y persistentes a lo largo del tiempo. Aunque pueden ocurrir cambios menores debido al envejecimiento, lesiones o enfermedades de la piel, la estructura general de las huellas dactilares se mantiene constante. Esto permite utilizar las huellas dactilares como un método confiable de identificación a lo largo de la vida de una persona.
- **Diversidad:** Las huellas dactilares varían significativamente entre diferentes personas, pero también dentro de una misma persona en sus diferentes dedos. Cada dedo tiene su propio patrón de huella dactilar, lo que aumenta la diversidad y la unicidad del sistema de identificación biométrica basado en huellas dactilares.
- **Facilidad de adquisición:** Las huellas dactilares pueden ser fácilmente adquiridas mediante métodos no invasivos y rápidos, como la toma de imágenes con sensores biométricos ópticos, capacitivos o ultrasonidos. Esto hace que la captura de huellas dactilares sea conveniente y accesible para su uso en sistemas de identificación y autenticación.
- **Resistencia a falsificaciones:** Las huellas dactilares son difíciles de falsificar o replicar debido a su complejidad y características únicas. Los sistemas biométricos de identificación de huellas dactilares suelen incorporar medidas de seguridad adicionales, como la detección de huellas vivas, para prevenir el uso de huellas falsas o copiadas.

En conjunto, estos fundamentos hacen que la identificación de huellas dactilares sea un método altamente confiable, seguro y ampliamente utilizado en una variedad de aplicaciones, desde la seguridad en dispositivos móviles hasta la investigación forense y la gestión de identidades.

9.1.2 Funciones de Huellas Dactilares

Las huellas dactilares cumplen varias funciones globales en el contexto de la identificación biométrica y la seguridad. Aquí hay algunas de esas funciones:

- **Identificación personal:** La función más evidente de las huellas dactilares es la identificación personal. Cada individuo tiene un conjunto único de huellas dactilares que se utiliza para autenticar su identidad en diversos sistemas, como el acceso a dispositivos móviles, la entrada a edificios seguros, la

verificación de identidad en aeropuertos y la gestión de registros médicos y financieros.

- **Control de acceso:** Las huellas dactilares se utilizan ampliamente en sistemas de control de acceso para garantizar la seguridad en áreas restringidas. Esto incluye el acceso a oficinas, laboratorios, instalaciones industriales, centros de datos y cualquier otro lugar donde se requiera una verificación precisa de la identidad de las personas que ingresan.
- **Prevención de fraudes:** Debido a su unicidad y dificultad para ser falsificadas, las huellas dactilares son una herramienta efectiva para prevenir fraudes. Se utilizan en sistemas de pagos biométricos, verificación de identidad en transacciones financieras y protección de información confidencial para garantizar que solo las personas autorizadas tengan acceso.
- **Seguridad informática:** En el ámbito de la seguridad informática, las huellas dactilares se utilizan para proteger dispositivos móviles, computadoras portátiles y sistemas de información. Los sensores de huellas dactilares integrados en estos dispositivos permiten a los usuarios autenticarse de manera segura y rápida sin necesidad de contraseñas o códigos de acceso.
- **Identificación forense:** En el campo de la investigación criminal, las huellas dactilares son una herramienta invaluable para la identificación de personas involucradas en delitos. Se utilizan para comparar y vincular huellas encontradas en la escena del crimen con las de sospechosos y personas en bases de datos forenses.
- **Gestión de identidades:** Las huellas dactilares se utilizan en la gestión de identidades en diversos contextos, como la emisión de documentos de identidad, la autenticación en sistemas gubernamentales y la verificación de identidad en servicios públicos y privados, garantizando la integridad y seguridad de la información personal.

9.1.3 Patrones de Huellas Dactilares

Los patrones de huellas dactilares se refieren a las características distintivas que forman los diseños únicos presentes en la superficie de la piel de los dedos de las manos y, en menor medida, de los pies. Estos patrones se clasifican en varios tipos principales, cada uno con características específicas que los distinguen:

1. Arco (o arco simple): Este patrón se caracteriza por líneas curvas que van de un lado del dedo al otro, formando un arco suave. En este tipo de patrón, no hay deltas (puntos de inicio) ni núcleos (centros), y es menos común que otros patrones.
2. Presilla (o lazo): En este patrón, las líneas forman bucles que se abren hacia el borde del dedo, creando una estructura de lazo. Se distingue por tener uno o más deltas (puntos de inicio) y uno o más núcleos (centros) dentro del bucle.
3. Espiral: Este patrón se caracteriza por líneas curvas que forman espirales o vueltas en la superficie de la piel. Es menos común que otros patrones y suele encontrarse en la parte interna de los dedos.
4. Compuesto: Este patrón combina características de los patrones de arco, presilla y espiral en una misma huella. Puede tener múltiples deltas, núcleos y estructuras de bucles.
5. Whorl (o verticilo): Este patrón se caracteriza por líneas que forman círculos concéntricos, espirales o patrones circulares alrededor de un núcleo central. Puede tener uno o más núcleos y deltas.

Adicionalmente, se pueden distinguir subtipos dentro de cada patrón principal, como los arcos dobles, las presillas ulnare o radiale, los espirales simples o dobles, y los whorls internos o externos, entre otros. Estos subtipos se definen por la disposición y la dirección de las líneas y bucles en la huella dactilar.

La combinación única de estos patrones, junto con otras características como bifurcaciones, terminaciones, puntos característicos y detalles menores, forma el perfil único de cada huella dactilar, que se utiliza para la identificación biométrica y la autenticación de identidades en sistemas de seguridad y control de acceso.

9.1.4 Huellas Dactilares Distintivas y Persistentes

las huellas dactilares son realmente distintivas y persistentes en cada individuo, y esto se debe a varias razones:

Unicidad: Cada persona tiene un conjunto único de huellas dactilares. Incluso entre gemelos idénticos, las huellas dactilares son diferentes. Esto se debe a la combinación única de patrones de crestas, surcos, bifurcaciones, terminaciones y otros detalles que se forman en la piel de los dedos durante el desarrollo fetal. La probabilidad de dos

personas teniendo exactamente las mismas huellas dactilares es extremadamente baja, lo que hace que las huellas sean una herramienta de identificación confiable.

Estabilidad: Las características de las huellas dactilares son estables y persistentes a lo largo del tiempo. Aunque pueden ocurrir cambios menores debido al envejecimiento, lesiones o enfermedades de la piel, la estructura general de las huellas dactilares se mantiene constante. Esto significa que las huellas dactilares son confiables para la identificación a lo largo de la vida de una persona.

Diversidad: Las huellas dactilares varían significativamente entre diferentes personas, pero también dentro de una misma persona en sus diferentes dedos. Cada dedo tiene su propio patrón de huella dactilar, lo que aumenta la diversidad y la unicidad del sistema de identificación biométrica basado en huellas dactilares.

Resistencia a falsificaciones: Las huellas dactilares son difíciles de falsificar o replicar debido a su complejidad y características únicas. Los sistemas biométricos de identificación de huellas dactilares suelen incorporar medidas de seguridad adicionales, como la detección de huellas vivas, para prevenir el uso de huellas falsas o copiadas.

9.2 ingeniería Inversa

La ingeniería inversa es un proceso mediante el cual se analiza un objeto o sistema complejo para comprender su funcionamiento interno o su diseño original. Esto se hace desmontando, examinando y estudiando sus componentes, estructura y comportamiento para obtener información detallada sobre cómo fue creado o cómo funciona.

En el contexto de la ingeniería de software, la ingeniería inversa se utiliza para entender el código fuente de un programa sin tener acceso a su documentación o diseño original. También se puede aplicar en campos como la ingeniería mecánica, electrónica e incluso en el estudio de productos comerciales para entender su tecnología o procesos de fabricación.

La ingeniería inversa es una herramienta poderosa que permite entender el funcionamiento interno y el diseño de objetos, sistemas y tecnologías complejas, siendo útil para una variedad de propósitos que van desde la mejora de productos hasta la seguridad y el desarrollo de nuevas tecnologías.

9.2.1 Decompiladores

Los decompiladores son herramientas de software que se utilizan en el contexto de la ingeniería inversa de programas informáticos. Su función principal es revertir el proceso de compilación, es decir, transformar el código ejecutable de un programa en un formato legible y comprensible para los humanos, como código fuente en un lenguaje de programación de alto nivel.

Cuando un programa se compila, su código fuente, escrito en un lenguaje de programación como C++, Java o C#, se traduce a un lenguaje de máquina que la computadora puede entender y ejecutar directamente. Esta transformación hace que el código fuente original sea difícil de recuperar solo a partir del ejecutable.

Los decompiladores trabajan de manera inversa a los compiladores. Analizan el código de la aplicación ejecutable y tratan de generar un equivalente aproximado en un lenguaje de programación de alto nivel. Sin embargo, es importante tener en cuenta que el código resultante de un decompilador puede no ser idéntico al código fuente original, ya que parte de la información sobre la estructura y el diseño del programa se pierde durante la compilación.

Los decompiladores son útiles en situaciones donde se necesita comprender el funcionamiento interno de un programa, estudiar su lógica de programación o realizar ingeniería inversa para fines legítimos, como la corrección de errores o la compatibilidad con sistemas diferentes. Sin embargo, es importante mencionar que su uso también puede plantear cuestiones éticas y legales, especialmente cuando se emplean para analizar programas protegidos por derechos de autor o para eludir medidas de seguridad.

9.2.2 Archivos .DLL

Los archivos DLL (Dynamic Link Libraries) son componentes de software en sistemas Windows que contienen código y datos que pueden ser utilizados por múltiples programas al mismo tiempo. Estos archivos son fundamentales para la organización y funcionamiento de muchas aplicaciones en el entorno Windows.

- **Funcionamiento compartido:** Una de las principales características de los archivos DLL es su capacidad para proporcionar funcionalidades compartidas a múltiples programas. Esto significa que varias aplicaciones pueden utilizar el mismo archivo DLL para acceder a rutinas, funciones y datos comunes, lo que reduce el tamaño de los programas individuales y promueve la reutilización de código.
- **Modularidad:** Los archivos DLL permiten el modularidad en el desarrollo de software. Esto significa que los desarrolladores pueden dividir sus programas en diferentes componentes, colocando funciones relacionadas en archivos DLL separados. Esto facilita la gestión y actualización de partes específicas de una aplicación sin afectar otras partes.
- **Optimización de recursos:** Al utilizar archivos DLL, las aplicaciones pueden optimizar el uso de recursos en el sistema, ya que solo cargan en memoria las partes del archivo DLL que necesitan en un momento dado. Esto puede mejorar el rendimiento general del sistema al reducir la carga de memoria y la duplicación de código.
- **Actualizaciones y correcciones:** Los archivos DLL también facilitan la distribución de actualizaciones y correcciones de errores. Al actualizar un archivo DLL compartido, todas las aplicaciones que dependen de él pueden beneficiarse de las mejoras sin necesidad de modificar cada aplicación por separado.
- **Dependencias:** Sin embargo, es importante tener en cuenta que las aplicaciones que utilizan archivos DLL también dependen de la disponibilidad y la versión correcta de esos archivos. Si un archivo DLL necesario no está presente o está dañado, puede causar errores o problemas de funcionamiento en las aplicaciones que lo requieren.

10. Herramientas Usadas

A continuación, se describen la serie de herramientas, desde hardware hasta software, utilizadas para realización de este proyecto.

10.1 Herramientas Físicas (hardware)

Componen el conjunto de herramientas físicas que se utilizaron en conjunto con otros materiales informáticos para la correcta realización del proyecto.

10.1.1 Lector Biométrico U.are.U 4500 Reader



El lector biométrico U.are.U 4500 Reader de Digital Persona es un dispositivo diseñado para la captura precisa y rápida de huellas dactilares con fines de identificación y autenticación biométrica. Digital Persona, una empresa líder en soluciones de autenticación biométrica, ha desarrollado este lector para ofrecer un alto nivel de seguridad y precisión en una variedad de aplicaciones, desde control de acceso hasta gestión de identidad.

Este lector biométrico se distingue por su diseño compacto y ergonómico, que lo hace fácil de usar y transportar. Utiliza tecnología óptica para escanear y capturar imágenes de alta resolución de las huellas dactilares, lo que garantiza una excelente calidad de imagen incluso en condiciones adversas, como huellas húmedas o secas, sucias o con cicatrices.

El U.are.U 4500 Reader se conecta a través de un puerto USB a dispositivos compatibles, como computadoras personales, terminales de punto de venta (POS), o sistemas de control de acceso. Su interfaz de usuario es intuitiva y fácil de entender, lo que facilita su integración y uso en diferentes entornos y aplicaciones.

Una de las características destacadas de este lector biométrico es su capacidad para realizar una autenticación rápida y precisa en tiempo real. Utiliza algoritmos avanzados de coincidencia de huellas dactilares para verificar la identidad de un individuo en cuestión de segundos, lo que lo hace ideal para aplicaciones donde la velocidad es esencial, como el control de acceso en edificios o la identificación en entornos de atención médica.

Además, el U.are.U 4500 Reader cuenta con características de seguridad adicionales, como cifrado de datos y protección contra manipulaciones, que garantizan la integridad y confidencialidad de la información biométrica almacenada y transmitida.

Decidimos utilizar el lector biométrico U.are.U 4500 Reader de DigitalPersona porque es una solución confiable y efectiva para la captura y verificación de huellas dactilares en una amplia gama de aplicaciones. Su combinación de precisión, velocidad y seguridad lo convierte en una herramienta valiosa para la autenticación biométrica en entornos comerciales, gubernamentales y de seguridad.

10.2 Herramientas Digitales (software)

Componen la serie de herramientas digitales que, por medio del uso de estas, somos capaces de interactuar con los elementos del hardware para el desarrollo y el uso de las pruebas del proyecto.

10.2.1 Visual Basic

Se ha utilizado Visual Basic como lenguaje de origen en donde programamos todos los métodos necesarios para el funcionamiento del lector biométrico.

Visual Basic (VB) es un lenguaje de programación de alto nivel desarrollado por Microsoft. Se introdujo por primera vez en 1991 como una parte integral de la suite de desarrollo de software de Microsoft, conocida como Microsoft Visual Studio. Desde entonces, ha experimentado varias iteraciones y versiones, siendo Visual Basic .NET (VB.NET) la versión más reciente y más utilizada en la actualidad.

Visual Basic es un lenguaje de programación poderoso y fácil de aprender que se utiliza ampliamente en el desarrollo de aplicaciones de escritorio, aplicaciones web y servicios empresariales. Su enfoque en la simplicidad y la productividad lo hace ideal para desarrolladores principiantes y profesionales que buscan crear aplicaciones de forma rápida y eficiente.

Algunas de las características y ventajas de Visual Basic incluyen:

- **Entorno de desarrollo integrado (IDE):** Visual Basic se utiliza típicamente con el entorno de desarrollo integrado (IDE) proporcionado por Microsoft Visual Studio. Este entorno ofrece herramientas avanzadas de desarrollo, depuración y diseño de interfaces gráficas de usuario, lo que facilita el proceso de desarrollo de aplicaciones.
- **Programación orientada a eventos:** En Visual Basic, la lógica de la aplicación se define en respuesta a eventos específicos, como clics de botones o cambios en los valores de los controles de la interfaz de usuario. Esto simplifica la estructura del código y facilita la comprensión del flujo de la aplicación.
- **Interfaz gráfica de usuario (GUI) fácil de crear:** Visual Basic proporciona un conjunto de herramientas y controles gráficos que permiten a los desarrolladores crear interfaces de usuario intuitivas y atractivas con facilidad. Estos controles pueden ser arrastrados y soltados en el diseño de formularios, lo que simplifica la creación de la interfaz de usuario.
- **Bibliotecas de clases y funciones integradas:** Visual Basic incluye una amplia gama de bibliotecas de clases y funciones integradas que permiten a los desarrolladores realizar una variedad de tareas comunes, como manipulación de archivos, acceso a bases de datos y gestión de errores, sin necesidad de escribir código desde cero.
- **Compatibilidad con el framework .NET:** Con la introducción de Visual Basic .NET, el lenguaje se ha integrado completamente con el framework .NET de Microsoft. Esto proporciona acceso a un amplio ecosistema de bibliotecas y servicios, así como interoperabilidad con otros lenguajes compatibles con .NET.

Ventajas

- Posee una curva de aprendizaje muy rápida.
- Integra el diseño e implementación de formularios de Windows.

- Permite usar con facilidad la plataforma de los sistemas Windows, dado que tiene acceso prácticamente total a la API de Windows, incluidas librerías actuales.
- Es uno de los lenguajes de uso más extendido, por lo que resulta fácil encontrar información, documentación y fuentes para los proyectos.
- Fácilmente extensible mediante librerías DLL y componentes ActiveX de otros lenguajes.

Desventajas

- Problema de versionado asociado con varias librerías runtime DLL, conocido como DLL Hell
- Soporte pobre para programación orientada a objetos¹⁰
- Incapacidad para crear aplicaciones multihilo, sin tener que recurrir a llamadas de la API de Windows.
- Dependencia de complejas y frágiles entradas de registro COM¹¹
- La capacidad de utilizar controles en un único formulario es muy limitada en comparación a otras herramientas.¹²

10.2.2 C#

Se ha utilizado C# en el proceso de decompilación de las bibliotecas dll. Además de que es lenguaje en el que DotPeek trabaja y nos permite ver los métodos y básicamente, todo el código de la biblioteca dll que genera visual Basic.

Ventajas

- Lenguaje moderno y versátil: C# es un lenguaje moderno que combina características de lenguajes como C++ y Java, pero con mejoras y simplificaciones en la sintaxis y funcionalidades.
- Integración con el ecosistema de Microsoft: Al ser desarrollado por Microsoft, C# se integra de manera natural con el ecosistema de herramientas y tecnologías de Microsoft, como Visual Studio (el entorno de desarrollo

integrado), .NET Framework y .NET Core (plataformas de desarrollo), Azure (plataforma en la nube), entre otros.

- Orientación a objetos: C# es un lenguaje orientado a objetos, lo que significa que se basa en conceptos como clases, objetos, herencia, polimorfismo, encapsulamiento, entre otros. reutilizables y mantenibles.
- Seguridad y control de tipos: C# es un lenguaje fuertemente tipado, lo que significa que proporciona un alto nivel de seguridad y control sobre los tipos de datos utilizados en el código.

Desventajas

- Plataforma específica: Aunque C# es compatible con diferentes sistemas operativos a través de .NET Core (y ahora .NET 5+), tradicionalmente ha estado más asociado con el entorno Windows.
- Curva de aprendizaje inicial: Aunque C# es considerado un lenguaje relativamente fácil de aprender, especialmente para quienes tienen experiencia previa en programación, aún puede tener una curva de aprendizaje para quienes se están iniciando en la programación o vienen de otros lenguajes con sintaxis y paradigmas diferentes.
- Dependencia del ecosistema de Microsoft: Si bien la integración con el ecosistema de Microsoft puede ser una ventaja, también puede ser una desventaja para quienes prefieren un enfoque más independiente o que trabajan en entornos donde las tecnologías y herramientas de Microsoft no son la primera opción.
- Performance en algunas áreas específicas: Aunque C# es un lenguaje de alto nivel y ofrece buen rendimiento en la mayoría de los casos, en ciertas áreas específicas como procesamiento intensivo de datos o aplicaciones con requerimientos muy críticos de rendimiento en tiempo real, pueden existir lenguajes o tecnologías que ofrezcan un rendimiento aún mejor.

10.2.3 Bibliotecas DLL

Se han utilizado las bibliotecas DLL como un método de comunicación entre VisualBasic y Python, exportamos nuestro proyecto de VisualBasic a una biblioteca DLL la cual podremos utilizar todos sus métodos en Python mediante Python.NET.

Las bibliotecas DLL (Dynamic Link Libraries) son archivos que contienen código y datos que pueden ser utilizados por múltiples programas simultáneamente en un sistema operativo Windows. Estas bibliotecas permiten compartir funciones y recursos entre diferentes aplicaciones, lo que contribuye a la eficiencia y modularidad del software.

Las DLL se llaman "bibliotecas dinámicas" porque el código que contienen se carga en memoria solo cuando se necesita durante la ejecución de un programa, en lugar de ser cargado en memoria al inicio como sucede con las bibliotecas estáticas. Esto permite ahorrar recursos de memoria, ya que las DLL se pueden compartir entre varios programas en ejecución.

En el contexto de aplicaciones en lenguajes como C, C++, y otros, las DLL son una parte fundamental del desarrollo de software. También son comunes en sistemas operativos Windows, donde muchos componentes del sistema y aplicaciones de terceros se implementan como DLL.

Cuando una aplicación utiliza una DLL, el sistema operativo carga la biblioteca en memoria y resuelve las llamadas a funciones en tiempo de ejecución. Este enfoque modular y compartido es una práctica común en el desarrollo de software para mejorar la eficiencia y facilitar el mantenimiento.

Las bibliotecas DLL se utilizan en una amplia variedad de aplicaciones, desde programas de usuario hasta sistemas operativos y controladores de dispositivos. Pueden contener funciones para realizar tareas como manipulación de archivos, acceso a bases de datos, gráficos, redes, entre otros. Además, muchas bibliotecas de terceros y sistemas operativos proporcionan funciones útiles a través de DLL que pueden ser utilizadas por los desarrolladores para extender las capacidades de sus aplicaciones.

10.2.4 HTML

HTML, o HyperText Markup Language, es el lenguaje estándar utilizado para crear y diseñar páginas web. Es esencial para la creación de páginas web y proporciona una base sólida para el desarrollo web, aunque su uso efectivo requiere la combinación con otros lenguajes y tecnologías para obtener resultados completos y moderno.

Se ha utilizado HTML como lenguaje para el desarrollo de las plantillas de las páginas web que se diseñaron en Django.

Ventajas

- **Facilidad de aprendizaje:** HTML es relativamente fácil de aprender, especialmente para quienes se inician en el desarrollo web.
- **Compatibilidad:** Es compatible con la mayoría de los navegadores web modernos, lo que garantiza que las páginas HTML se puedan ver correctamente en diferentes dispositivos.
- **Flexibilidad:** Permite integrar otros lenguajes como CSS para estilos y JavaScript para interactividad, lo que amplía sus capacidades y posibilidades de diseño.
- **Estándar universal:** Es un estándar reconocido y ampliamente utilizado en la industria del desarrollo web, lo que facilita la colaboración y la creación de contenido web consistente.
- **SEO y accesibilidad:** HTML5 introduce elementos semánticos que mejoran la accesibilidad y la optimización para motores de búsqueda al proporcionar una estructura más significativa y clara del contenido.

Desventajas:

- **Limitaciones en diseño:** HTML por sí solo tiene limitaciones en términos de diseño y presentación visual avanzada. Para estilos más complejos, se requiere CSS.
- **Interactividad limitada:** Para funcionalidades interactivas más avanzadas, como formularios dinámicos o efectos visuales, es necesario usar JavaScript u otras tecnologías.
- **Dependencia de otros lenguajes:** Para crear páginas web completas y modernas, a menudo es necesario combinar HTML con CSS para estilos y JavaScript para interactividad, lo que puede aumentar la complejidad del desarrollo.
- **Compatibilidad con versiones antiguas:** Aunque HTML5 ha mejorado muchas áreas, la compatibilidad con versiones anteriores de HTML y navegadores obsoletos aún puede ser un desafío en algunos casos.

10.2.5 Bootstrap

Bootstrap es un framework de diseño front-end desarrollado por Twitter que proporciona un conjunto de herramientas y estilos predefinidos para crear sitios web responsivos y atractivos.

Se ha utilizado Bootstrap para implementarlo en las plantillas HTML de las páginas web. Por medio del uso de este framework se pueden conseguir resultados más rápidos y notables en menos tiempo.

Ventajas:

- **Facilidad de uso:** Bootstrap simplifica el proceso de diseño web al proporcionar componentes predefinidos y estilos listos para usar.
- **Responsividad:** Los sitios creados con Bootstrap son responsivos por defecto, lo que significa que se ajustan automáticamente al tamaño de la pantalla del dispositivo, mejorando la experiencia del usuario en dispositivos móviles y de escritorio.
- **Compatibilidad:** Bootstrap es compatible con la mayoría de los navegadores modernos, lo que garantiza una experiencia consistente para los usuarios.
- **Documentación completa:** Bootstrap cuenta con una documentación extensa y detallada, que incluye ejemplos y guías para ayudar a los desarrolladores a utilizar eficazmente todas sus características.
- **Personalización:** Aunque Bootstrap ofrece estilos y componentes predefinidos, también permite una gran personalización a través de variables CSS y clases adicionales, lo que permite adaptarse a las necesidades específicas de diseño de cada proyecto.

Desventajas:

- **Pesadez:** Al ser un framework completo con muchas características y estilos, el uso completo de Bootstrap puede resultar en archivos CSS y JavaScript

relativamente grandes, lo que puede afectar el rendimiento del sitio web si no se optimiza adecuadamente.

- Sobrecarga de estilos: A veces, el uso excesivo de estilos predefinidos de Bootstrap puede hacer que todos los sitios creados con este framework se vean similares, a menos que se realicen personalizaciones significativas.
- Curva de aprendizaje inicial: Aunque Bootstrap es fácil de usar para desarrolladores con experiencia en CSS y HTML, puede tener una curva de aprendizaje inicial para quienes son nuevos en el desarrollo web.
- Dependencia externa: Siempre que se utilice Bootstrap, se crea una dependencia externa en el framework, lo que significa que cualquier cambio o actualización en Bootstrap puede afectar el funcionamiento y la apariencia del sitio web.

10.2.4 Python

Python es un lenguaje de programación de alto nivel, interpretado y de propósito general. Fue creado por Guido van Rossum y lanzado por primera vez en 1991. Se caracteriza por su sintaxis clara y legible, que hace que sea fácil de aprender y entender, incluso para aquellos que están comenzando en la programación.

Python es conocido por su enfoque en la legibilidad del código y su filosofía de "Zen de Python", que promueve la simplicidad y la claridad en el diseño del código. Esto lo hace especialmente popular entre los principiantes y los desarrolladores que priorizan la facilidad de uso y la productividad.

Una de las fortalezas de Python es su amplia biblioteca estándar, que incluye módulos y funciones para una variedad de tareas, desde manipulación de archivos hasta desarrollo web y análisis de datos. Además, cuenta con una comunidad activa que contribuye con una amplia gama de bibliotecas y frameworks de código abierto que extienden sus capacidades y lo hacen adecuado para una variedad de aplicaciones.

Python es multiplataforma, lo que significa que puede ejecutarse en diferentes sistemas operativos, como Windows, macOS y Linux, sin modificaciones

significativas. Esto lo convierte en una opción versátil para el desarrollo de software en una variedad de entornos.

Características y paradigmas

Python es un lenguaje de programación multiparadigma. Esto significa que más que forzar a los programadores a adoptar un estilo particular de programación, permite varios estilos: programación orientada a objetos, programación imperativa y programación funcional. Otros paradigmas están soportados mediante el uso de extensiones.

Python usa tipado dinámico y conteo de referencias para la gestión de memoria.

Una característica importante de Python es la resolución dinámica de nombres; es decir, lo que enlaza un método y un nombre de variable durante la ejecución del programa (también llamado enlace dinámico de métodos).

Otro objetivo del diseño del lenguaje es la facilidad de extensión. Se pueden escribir nuevos módulos fácilmente en C o C++. Python puede incluirse en aplicaciones que necesitan una interfaz programable.²⁶

Aunque la programación en Python podría considerarse en algunas situaciones hostil a la programación funcional tradicional expuesta por Lisp, existen bastantes analogías entre Python y los lenguajes minimalistas de la familia Lisp (como Scheme).

Python fue utilizado como lenguaje de programación en el cual fue montada la interfaz del lector biométrico por medio de el framework Django, para hacer las conexiones con los métodos en VisualBasic se utilizaron bibliotecas .dll y el paquete pythonnet (CLR) que permite una comunicación e incrustación con .NET.

10.2.5 Python.NET

Python.NET (pythonnet) es un paquete que brinda a los programadores de Python una integración casi perfecta con .NET 4.0+ Common Language Runtime (CLR) en Windows y Mono runtime en Linux y OSX. Python.NET proporciona una poderosa herramienta de secuencias de comandos de aplicaciones para desarrolladores de .NET. Con este paquete, puede crear secuencias de comandos de aplicaciones .NET o crear

aplicaciones completas en Python, utilizando servicios y componentes .NET escritos en cualquier lenguaje destinado a CLR (C#, VB.NET, F#, C++/CLI).

Tenga en cuenta que este paquete `_no_` implementa Python como lenguaje CLR de primera clase; no produce código administrado (IL) a partir del código Python. Más bien, es una integración del motor CPython con el tiempo de ejecución .NET o Mono. Este enfoque le permite utilizar servicios CLR y continuar usando el código Python existente y las extensiones C-API mientras mantiene las velocidades de ejecución nativas del código Python. Si está interesado en una implementación de código administrado puro del lenguaje Python, debería consultar el proyecto IronPython .

Actualmente, Python.NET es compatible y está probado con las versiones de Python desde la 3.7 en adelante.

Las versiones actuales están disponibles en PyPi y Nuget.org .

10.2.6 Visual Studio

Visual Studio es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) creado por Microsoft. Se utiliza para desarrollar aplicaciones de software para Windows, así como también para aplicaciones web, móviles y de escritorio. Visual Studio ofrece un conjunto de herramientas poderosas y funciones integradas que facilitan el proceso de desarrollo de software en diferentes plataformas y lenguajes de programación.

Algunas características de Visual Studio son:

- **Soporte multiplataforma:** Visual Studio admite el desarrollo de aplicaciones para una variedad de plataformas, incluyendo Windows, iOS, Android y Linux. Esto permite a los desarrolladores crear aplicaciones para una amplia gama de dispositivos y sistemas operativos.
- **Amplia gama de lenguajes de programación:** Visual Studio admite múltiples lenguajes de programación, incluyendo C#, Visual Basic, C++, F#, Python, JavaScript, TypeScript, y más. Esto permite a los desarrolladores elegir el lenguaje que mejor se adapte a sus necesidades y preferencias.
- **Herramientas de depuración avanzadas:** Visual Studio ofrece herramientas de depuración avanzadas que permiten a los desarrolladores identificar y corregir errores en su código de manera eficiente. Estas herramientas incluyen puntos de interrupción, inspección de variables, seguimiento de pila y más.

- **Diseñador de interfaz gráfica de usuario (GUI):** Visual Studio incluye un diseñador de GUI que permite a los desarrolladores crear interfaces de usuario intuitivas y atractivas para sus aplicaciones. Los desarrolladores pueden arrastrar y soltar controles en el diseñador para crear fácilmente la apariencia deseada de su aplicación.
- **Integración con servicios en la nube:** Visual Studio se integra con servicios en la nube como Azure, lo que permite a los desarrolladores crear, probar y implementar aplicaciones en la nube de manera fácil y rápida. Esto facilita el desarrollo de aplicaciones escalables y distribuidas en la nube.
- **Control de versiones:** Visual Studio incluye herramientas de control de versiones que permiten a los desarrolladores colaborar en proyectos de manera eficiente. Esto incluye integración con sistemas de control de versiones como Git y Team Foundation Version Control (TFVC).
- **Extensibilidad:** Visual Studio es altamente extensible y admite una amplia gama de extensiones y complementos desarrollados por la comunidad. Estas extensiones pueden agregar nuevas características y funcionalidades al IDE, personalizándolo para adaptarse a las necesidades específicas de cada desarrollador.

10.2.7 Visual Studio Code

Utilizamos VisualStudio Code principalmente para desarrollar nuestro código de Python. Visual Studio Code (VS Code) es un editor de código fuente desarrollado por Microsoft que se ha convertido en una herramienta popular entre los desarrolladores debido a su versatilidad, rendimiento y amplia gama de características. Aunque comparte el nombre con Visual Studio, es importante destacar que Visual Studio Code es un editor de código ligero y de propósito general, mientras que Visual Studio es un entorno de desarrollo integrado (IDE) más completo.

Visual Studio Code es un editor de código fuente construido sobre el framework Electron. Es compatible con varios lenguajes de programación y un conjunto de características que pueden o no estar disponibles para un lenguaje dado, como se muestra en la siguiente tabla. Muchas de las características de Visual Studio Code no están expuestas a través de los menús o la interfaz de usuario. Más bien, se accede a través de la paleta de comandos o a través de archivos .json (por ejemplo, preferencias del usuario). La paleta de comandos es una interfaz de línea de comandos. Sin embargo, desaparece si el usuario hace clic fuera de él o presiona una combinación de teclas en el teclado para interactuar con algo que está fuera de él. Esto también se

aplica a los comandos que requieren mucho tiempo. Cuando esto sucede, el comando en progreso se cancela.

En el rol de editor de código fuente, Visual Studio Code permite cambiar la página de códigos en la que se guarda el documento activo, el carácter que identifica el salto de línea (una opción entre LF y CRLF) y el lenguaje de programación del documento activo.

10.2.8 DotPeek

Utilizamos DotPeek para analizar las bibliotecas DLL creadas con VisualStudio y poder saber que métodos debíamos utilizar en VisualStudio Code en Python. dotPeek es una herramienta independiente gratuita basada en el decompilador incluido en ReSharper. Puede decompilar de forma segura cualquier ensamblado .NET a código equivalente C# o IL.

El decompilador admite varios formatos, incluidas bibliotecas (.dll), ejecutables (.exe), y archivos de metadatos de Windows (.winmd).

dotPeek es una herramienta de descompilación y exploración de código desarrollada por JetBrains. Está diseñada para permitir a los desarrolladores examinar el código de ensamblados .NET, incluidas bibliotecas, aplicaciones y complementos, para comprender cómo funcionan o para realizar ingeniería inversa en aplicaciones de terceros.

Características

Descompilación de código fuente: dotPeek puede descompilar ensamblados .NET y mostrar el código fuente original, incluyendo comentarios y nombres de variables cuando están disponibles. Esto permite a los desarrolladores ver el código detrás de las aplicaciones y bibliotecas .NET, lo que puede ser útil para comprender su funcionamiento interno.

Navegación de código: dotPeek proporciona herramientas de navegación avanzadas que permiten a los desarrolladores moverse fácilmente por el código descompilado.

Esto incluye la capacidad de buscar símbolos, clases, métodos y referencias dentro de un ensamblado, lo que facilita la comprensión de la estructura y la lógica del código.

Soporte para múltiples formatos de archivo: dotPeek puede abrir y descompilar una variedad de formatos de archivo .NET, incluyendo archivos DLL, EXE, y otros formatos de biblioteca y ensamblado. Esto le permite trabajar con una amplia gama de aplicaciones y bibliotecas .NET.

Explorador de ensamblados: dotPeek incluye un explorador de ensamblados que muestra una lista de todos los ensamblados cargados en el proyecto actual. Los desarrolladores pueden navegar por esta lista para seleccionar y abrir ensamblados específicos para su análisis.

Funcionalidad de búsqueda: dotPeek proporciona una función de búsqueda que permite a los desarrolladores buscar palabras clave específicas dentro del código descompilado. Esto facilita la ubicación de secciones relevantes de código y ayuda a los desarrolladores a encontrar rápidamente lo que están buscando.

Integración con ReSharper: Si tienes instalado ReSharper (otra herramienta de JetBrains para Visual Studio), dotPeek se integra con él para proporcionar una experiencia de desarrollo más completa y mejorada.

10.2.9 SDK Digital Persona

El SDK es una herramienta muy importante para poder realizar este proyecto, es un conjunto de librerías y recursos que permiten a las bibliotecas DLL de .NET comunicarse con el lector biométrico y realizar todas sus funciones. El SDK (kit de desarrollo de software, por sus siglas en inglés) de DigitalPersona para lector biométrico es un conjunto de herramientas, bibliotecas y recursos proporcionados por DigitalPersona, una empresa líder en tecnología biométrica, para permitir a los desarrolladores integrar la funcionalidad de los lectores biométricos de DigitalPersona en sus propias aplicaciones.

Este SDK proporciona una interfaz de programación de aplicaciones (API) que permite a los desarrolladores interactuar con los lectores biométricos de DigitalPersona desde sus aplicaciones de software. Esto incluye la capacidad de capturar y procesar huellas dactilares, autenticar usuarios basados en huellas dactilares y realizar otras operaciones biométricas.

El SDK permite a los desarrolladores capturar imágenes de huellas dactilares utilizando los lectores biométricos de DigitalPersona. Esto puede incluir funcionalidades como la detección de huellas dactilares válidas y la calidad de la imagen, otra característica es la extracción y comparación de características

biométricas, el SDK puede proporcionar funciones para extraer características biométricas únicas de las huellas dactilares capturadas y compararlas con otras huellas dactilares para realizar la autenticación de usuarios.

Algunos SDKs de DigitalPersona pueden incluir herramientas para gestionar usuarios, incluyendo el registro de nuevas huellas dactilares en la base de datos del sistema y la eliminación de huellas dactilares existentes.

El SDK puede incluir características para garantizar la seguridad y el cifrado de los datos biométricos capturados y procesados, protegiendo la privacidad y la integridad de la información biométrica. Dependiendo del SDK específico, puede estar disponible soporte para diferentes plataformas (como Windows, Linux, Android, etc.) y lenguajes de programación (como C#, Java, Python, etc.).

Productos de hardware compatibles con DigitalPersona

One Touch para Windows SDK: .NET Edition admite los siguientes productos de hardware de DigitalPersona:

- Lectores y módulos de huellas digitales DigitalPersona U.are.U 4000B/4500 o posteriores
- Compatibilidad de la plantilla de huellas dactilares del teclado de huellas dactilares DigitalPersona U.are.U

Las plantillas de huellas digitales producidas por todas las ediciones de One Touch para Windows SDK también son compatibles con los siguientes SDK de DigitalPersona:

- SDK de oro
- Oro CE SDK
- One Touch para Linux SDK, todas las distribuciones

10.2.10 Django

Django es un framework de desarrollo web de alto nivel y de código abierto, escrito en Python. Fue creado originalmente en 2003 por los desarrolladores web Adrian Holovaty y Simon Willison mientras trabajaban en el periódico Lawrence Journal-World en Kansas, Estados Unidos. Desde entonces, Django ha evolucionado y se ha convertido

en uno de los frameworks web más populares y ampliamente utilizados en la comunidad de desarrollo.

Aunque Django está fuertemente inspirado en la filosofía de desarrollo Modelo Vista Controlador, sus desarrolladores declaran públicamente que no se sienten especialmente atados a observar estrictamente ningún paradigma particular, y en cambio prefieren hacer "lo que les parece correcto". Como resultado, por ejemplo, lo que se llamaría "controlador" en un "verdadero" framework MVC se llama en Django "vista", y lo que se llamaría "vista" se llama "plantilla".

Gracias al poder de las capas mediator y foundation, Django permite que los desarrolladores se dediquen a construir los objetos Entity y la lógica de presentación y control para ellos.

Presentación

Aquí se maneja la interacción entre el usuario y el computador. En Django, esta tarea la realizan el motor de plantillas y el cargador de plantillas que toman la información y la presentan al usuario (vía HTML, por ejemplo). El sistema de configuración de URLs es también parte de la capa de presentación.

Control

En esta capa reside el programa o la lógica de aplicación en sí. En Django son representados por las vistas y los manipuladores. La capa de presentación depende de esta y a su vez esta depende de la capa de dominio.

Mediator

Es el encargado de manejar la interacción entre el subsistema Entity y foundation. Aquí se realiza el mapeo objeto-relacional a cargo del motor de Django.

Entity

El subsistema entity maneja los objetos de negocio. El mapeo objeto-relacional de Django permite escribir objetos de tipo entity de una forma fácil y estándar.

Foundation

La principal tarea del subsistema foundation es la de manejar a bajo nivel el trabajo con la base de datos. Se provee soporte a nivel de foundation para varias bases de datos y otras están en etapa de prueba.

Características

- **Productividad:** Django está diseñado para facilitar y acelerar el proceso de desarrollo web. Proporciona una amplia gama de componentes reutilizables y

funcionalidades integradas que permiten a los desarrolladores crear aplicaciones web complejas con un tiempo de desarrollo reducido.

- **Diseño basado en el modelo-vista-controlador (MVC):** Django sigue un patrón de diseño MVC, aunque utiliza una variante conocida como Modelo-Vista-Template (MVT). Este enfoque separa claramente la lógica de negocio (modelo), la presentación (template) y la lógica de la aplicación (view), lo que facilita la organización y mantenimiento del código.
- **ORM (Mapeo Objeto-Relacional) incorporado:** Django incluye su propio ORM, que permite a los desarrolladores interactuar con la base de datos utilizando objetos Python en lugar de SQL directamente. Esto simplifica el acceso y la manipulación de datos y hace que el código sea más legible y fácil de mantener.
- **Administrador de Django:** Django proporciona un potente y flexible sistema de administración de contenido que permite a los desarrolladores crear fácilmente interfaces de administración para sus aplicaciones web. Esto permite a los administradores del sitio gestionar contenido, usuarios y otros aspectos de la aplicación sin necesidad de escribir código adicional.
- **Seguridad integrada:** Django ofrece una serie de características integradas para garantizar la seguridad de las aplicaciones web, incluyendo protección contra ataques CSRF (Cross-Site Request Forgery), inyección de SQL y otros tipos de ataques comunes.
- **Escalabilidad y rendimiento:** Django está diseñado para ser escalable y eficiente en términos de rendimiento, lo que permite a las aplicaciones web manejar grandes volúmenes de tráfico y usuarios concurrentes sin problemas.

11. Desarrollo del Proyecto

El desarrollo del proyecto centrado en la experimentación y análisis para la detección y reconocimiento de huellas dactilares es un trabajo de gran importancia y complejidad en el ámbito de la seguridad y la tecnología. Las huellas dactilares son un medio de identificación único y ampliamente utilizado en la investigación criminal, la seguridad en accesos y en una variedad de aplicaciones biométricas.

El objetivo de este proyecto es diseñar y desarrollar un sistema avanzado que permita capturar, analizar y reconocer huellas dactilares con alta precisión y eficiencia. Esto implica el uso de tecnologías innovadoras en la captura de imágenes, procesamiento de señales, análisis de patrones y técnicas de aprendizaje automático para mejorar la exactitud y velocidad del reconocimiento biométrico.

11.1 Instalaciones

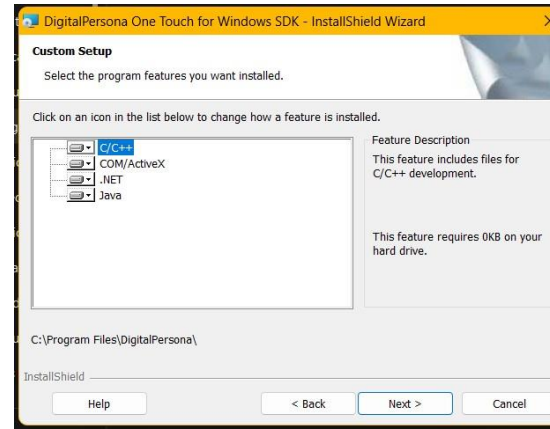
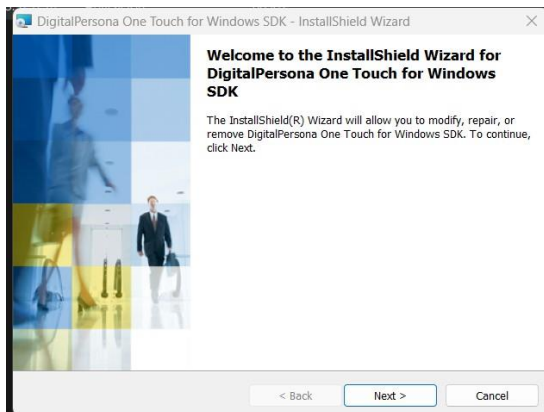
En esta sección se comenzará con la documentación referente al desarrollo del proyecto; la instalación de los diversos softwares que se utilizaron y el cómo estos fungieron de manera colaborativa para poder realizar el mismo.

Se documentará los siguientes puntos:

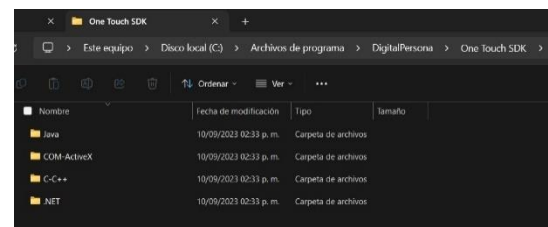
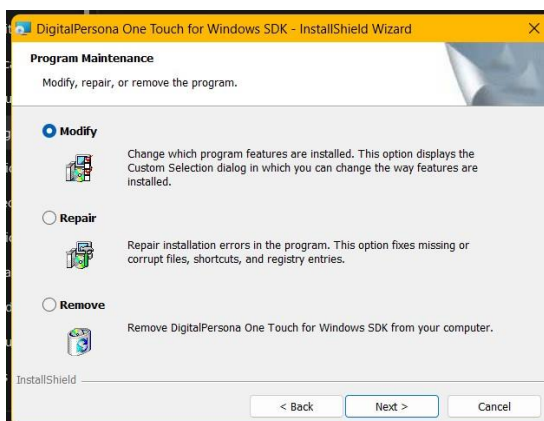
- Preparación del software: Todo lo necesario para que la instalación pueda terminar correctamente.
- Documentación de instalación: La documentación necesaria durante lo ocurrido en la instalación.
- Entorno de instalación: El entorno donde se va a llevar a cabo
- Proceso de instalación: Proceso mediante el cual se lleva a cabo y la selección de diversas opciones.
- Configuración y personalización: Dado que es posible que no vástes con una instalación básica, deberemos agregar información adicional o bien, disponer de ciertos conocimientos para mejorar la experiencia una vez terminada la instalación

11.1.1 Digital Persona SDK

Una vez descargado el paquete, damos doble clic y lo abrimos.



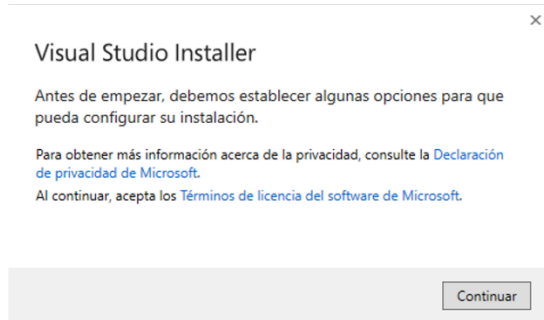
Daremos clic en modify o bien install según sea la primera o no vez que se esté instalando.



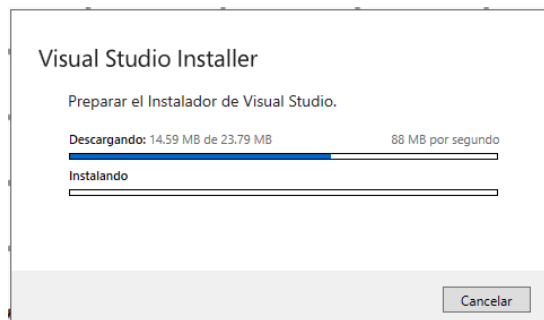
Daremos clic en siguiente, logrando así que se instale por defecto.

11.1.2 Visual Studio

Descargamos el instalador de la web oficial. Abrimos el programa con doble clic y daremos clic en continuar.

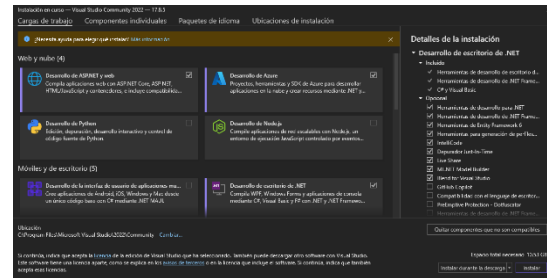


Se comenzaran a descargar varios archivos.



Después de complete la descarga daremos clic en finalizar e iniciar el programa, después nos mostrará una pantalla donde podemos seleccionar varias opciones o agregados para que se agreguen a la instalación de Visual Studio Code.

En esta sección solamente daremos clic en Desarrollo de Azurfe, .net y ARMET y web.

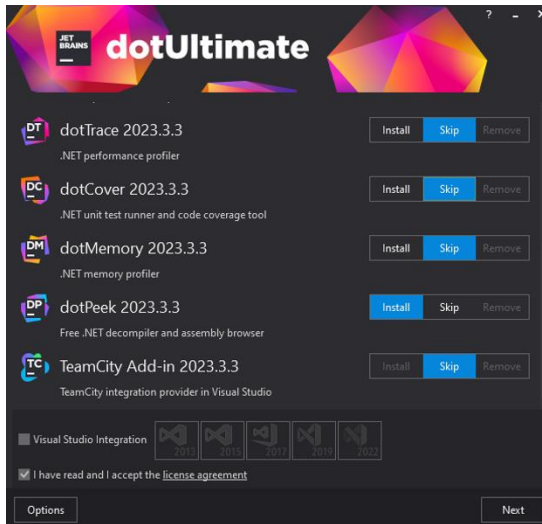


Cuando haya finalizado la descarga y la instalación de los archivos adicionales habrá terminado la instalación y podremos abrir Visual Studio Code sin problemas para empezar a trabajar con el.

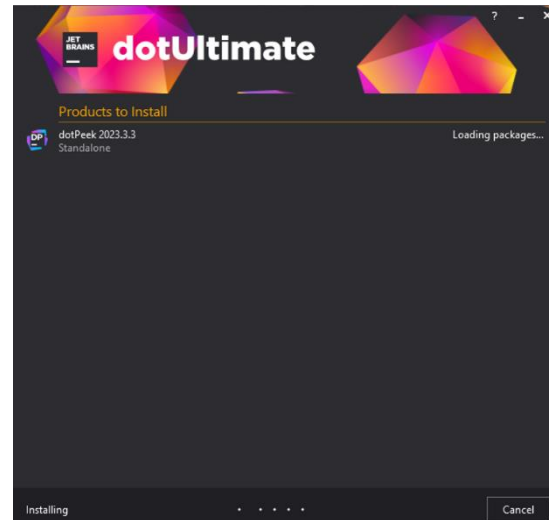
11.1.3 DotPeek

Vamos a descargar dotpeek de la web oficial y después daremos doble clic para abrir el instalador. Daremos en install.

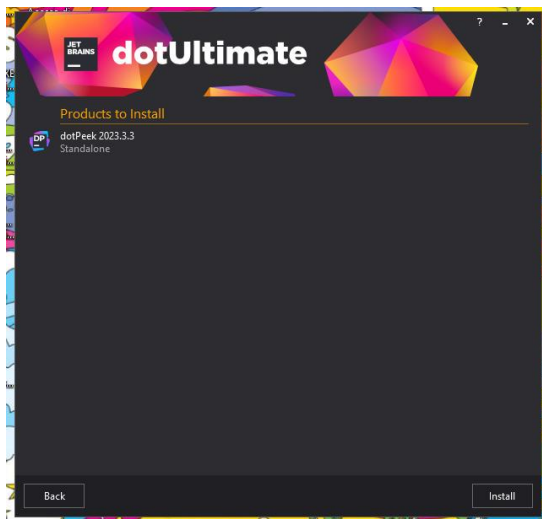
Esperemos hasta que termine el proceso de manera autonoma. Cuando haya terminado daremos clic en finalizar y se abra dotPeek.



Daremos clic en el botón “Install”.



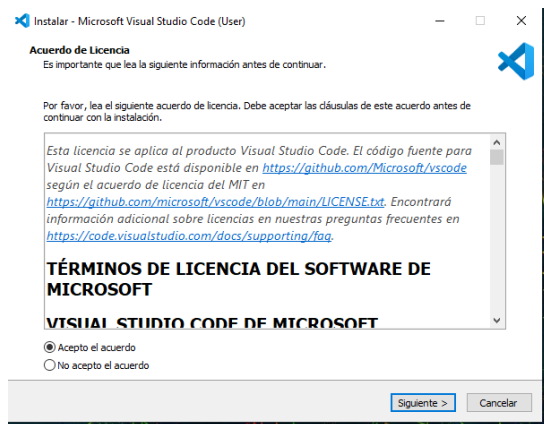
Comprobamos que todo salio bien abriendo un archivo del tipo .dll.



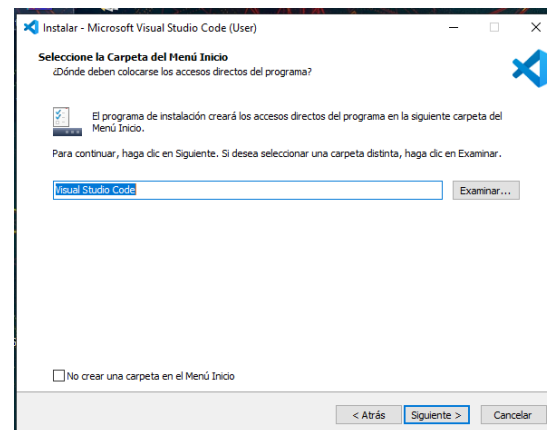
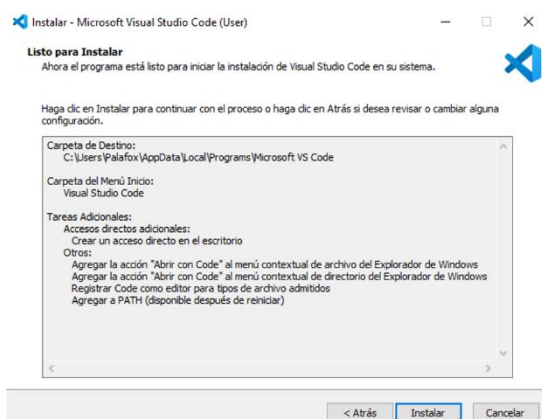
11.1.4 Visual Studio Code

Descargaremos visual studio code de la web oficial. Después abriremos el archivo con doble clic.

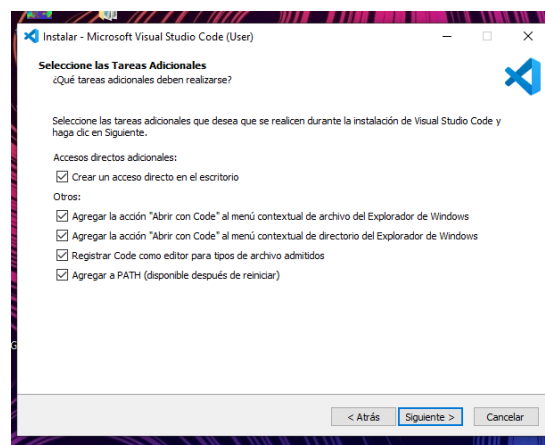
Daremos clic en siguiente. Nuevamente daremos clic en siguiente.



Cotinuiremos dando clic en siguiente.
Daremos clic en instalar.

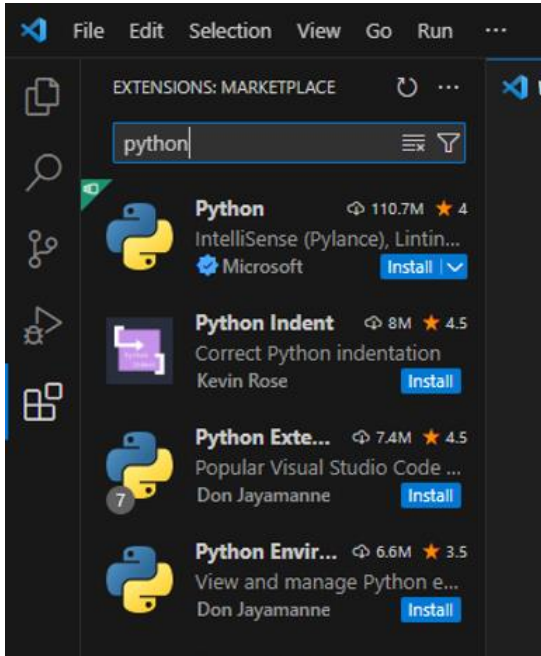


El proceso continuara de manera automatica hasta que termine. Finalmente daremos clic en finalizar para terminar el proceso



11.1.5 Python

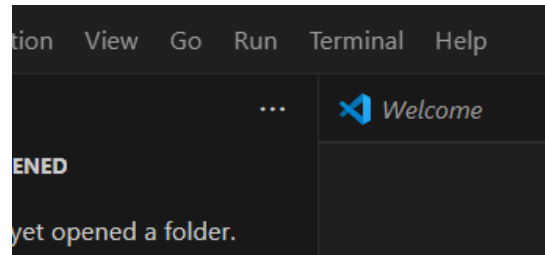
Para instalar Python es necesario abrir visual studio code. Una vez abierto hay que ir a la pestaña de extensiones y después buscar Python.



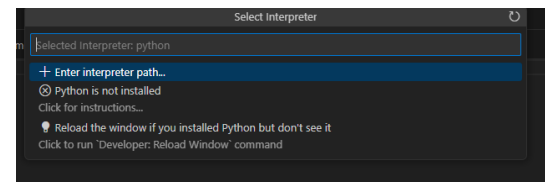
Daremos clic en install y nos mostrará la pantalla de la extensión. A continuación, comenzará la instalación de Python.



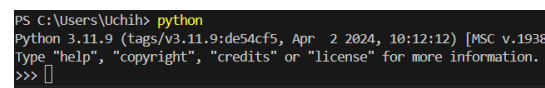
A continuación, iremos al menú view y luego en la daremos clic en la opción de paleta de comandos. Buscaremos: seleccionar interprete.



Una vez seleccionada esa opción nos mostrará una lista de versiones de Python. Seleccionamos la que hemos instalado (en caso de que haya más de una).



Comprobamos que la instalación fue un éxito.



11.1.6 Pythonnet

Para instalar Pythonnet necesitamos ir a la terminal de visual studio code. Después ingresar el siguiente código. El proceso terminará automáticamente.

```
PS C:\Users\Uchih> pip install pythonnet  
Requirement already satisfied: pythonnet in c:\  
211\site-packages (3.0.2)
```

El proceso continuara de manera automática hasta que descarguen todos los archivos y después se instalen.

Finalmente podremos comprobar que todo salió bien con el siguiente comando.

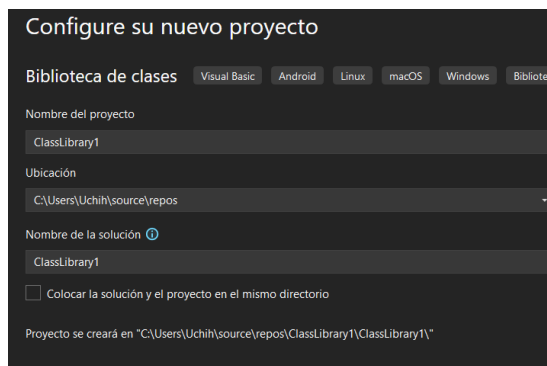
```
PS C:\Users\Uchih> python -c "import clr; print(clr.__version__)"  
>>  
3.0.2  
PS C:\Users\Uchih> █
```

11.2. Desarrollo del Proyecto

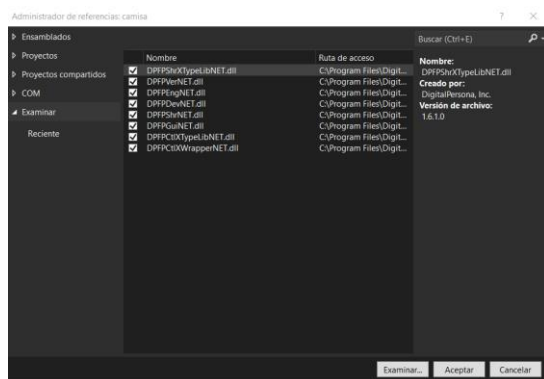
11.2.1 Biblioteca DLL

Lo primero que tenemos que hacer es crear un nuevo proyecto en Visual Studio 2022.

Seleccionaremos el tipo de proyecto como un proyecto de Biblioteca de Clases y tendrá que ser con el lenguaje de Visual Basic.

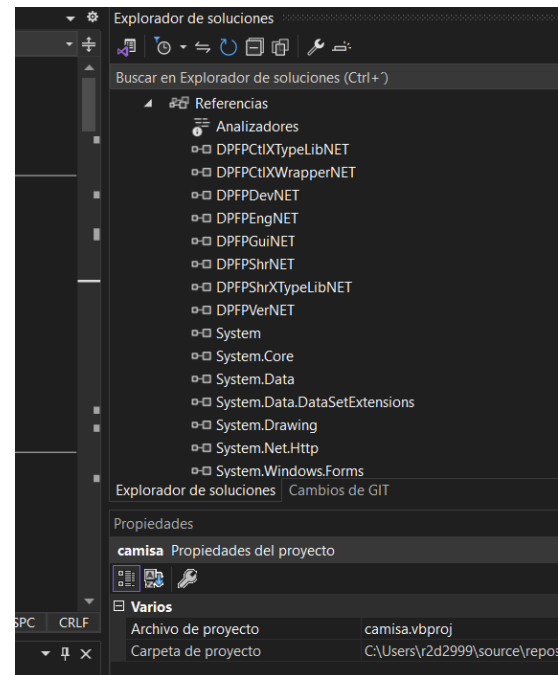


Cuando tengamos nuestro nuevo proyecto listo lo siguiente será agregar las referencias necesarias, en este caso utilizaremos los archivos .dll que se encuentran en la carpeta donde se instaló el SDK de Digital Persona.



Daremos clic en aceptar para agregar las referencias.

Cuando se hayan agregado las referencias bastará con ir al panel derecho del todo y dar clic en la parte de referencias para comprobar que estas se agregaron correctamente.



Lo siguiente será comenzar a escribir el código de la biblioteca que contendrá los métodos del SDK para poder ser utilizados por medio de Pythonnet en Python.

Crearemos una clase, que en este caso se ha nombrado Class1 y dentro de esa clase escribiremos todo el código que se va a ocupar.

Es necesario realizar los Imports correspondientes a las bibliotecas, tanto de SDK de Digital Persona, como a los propios de .net

Vamos a utilizar el EventHandler propio del SDK ya que ese método se encargará de crear un servicio que se ejecuta en segundo plano, esperando que se activen diferentes eventos según el sensor recopile los datos.

Se definen una serie de variables que se utilizaran en código posterior.

```
1 Imports DPFP
2 Imports DPFP.Capture
3 Imports System.Windows.Forms
4 Imports System.Text
5 Imports System.Drawing
6 Imports System.IO.MemoryMappedFiles
7
8 0 referencias
9 Public Class Class1
10 Implements DPFP.Capture.EventHandler
11
12 Public Captura As DPFP.Capture.Capture
13 Public Enroller As DPFP.Processing.Enrollment
14 Public Delegate Sub _delegadoMuestra(ByVal text As String)
15
16 Public mapas As New Bitmap(300, 300)
17
18 Public photonames As String = ""
19 Public listPaths As New List(Of String)
20
21
```

Lo siguiente a realizar es sobre escribir el método init() donde, por medio de un try and catch, intentaremos realizar una nueva captura (lo que permite que el sensor capture las huellas)

Vamos a asignarle el EventHandler a la variable de tipo capture y en caso de que no se posible, mandaremos los mensajes de errores correspondiente.

```
4 0 referencias
5 Public Overridable Sub Init()
6
7 Try
8     Captura = New Capture()
9
10 If Not Captura Is Nothing Then
11     Captura.EventHandler = Me
12     Enroller = New DPFP.Processing.Enrollment
13     Dim text As New StringBuilder()
14
15 Else
16     MessageBox.Show("No se pudo instanciar la captura")
17 End If
18
19 Catch ex As Exception
20     MessageBox.Show("No se pudo inicializar la captura")
21 End Try
22
23 End Sub
```

El método iniciar captura se encarga permitir que el sensor se inicialice, también hace uso del try and catch, para reportar cualquier tipo de error.

```
1 1 referencia
2 Public Sub IniciarCaptura()
3
4 If Not Captura Is Nothing Then
5
6 Try
7     Captura.StartCapture()
8
9 Catch ex As Exception
10     MessageBox.Show("no se pudo iniciar la captura")
11 End Try
12
13 End If
14
15 End Sub
```

El método paraCaptura() hace todo lo contrario al método anterior; detiene la captura, por ende, el funcionamiento del sensor.

Es de vital importancia, ya que de otro modo podría haber conflictos en el manejo de los hilos.

```
2 referencias
Public Sub paraCaptura()
    If Not Captura Is Nothing Then
        Try
            Captura.StopCapture()
        Catch ex As Exception
            MessageBox.Show("No se puede detener")
        End Try
    End If
End Sub

0 referencias
```

Los siguientes métodos también se sobre escriben nada más invocar el eventHandler.

El conjunto de estos métodos complementan los diferentes eventos que puedan ocurrir o no, cuando el sensor sea interactuado de diversas maneras.

```
1 Referencias
Public Sub OnFingerGone(Capture As Object, ReaderSerialNumber As String) Implements EventHandler.OnFingerGone
End Sub

2 Referencias
Public Sub OnFingerTouch(Capture As Object, ReaderSerialNumber As String) Implements EventHandler.OnFingerTouch
    MessageBox.Show("Lo tocaste!")
End Sub

3 Referencias
Public Sub OnHeaderConnect(Capture As Object, ReaderSerialNumber As String) Implements EventHandler.OnHeaderConnect
    MessageBox.Show("se ha conectado")
End Sub

4 Referencias
Public Sub OnHeaderDisconnect(Capture As Object, ReaderSerialNumber As String) Implements EventHandler.OnHeaderDisconnect
    MessageBox.Show("se ha desconectado")
End Sub

5 Referencias
Public Sub OnSampleQuality(Capture As Object, ReaderSerialNumber As String, CaptureFeedback As CaptureFeedback) Implements
End Sub
```

El método onComplete es un método que se sobre escribe cuando invocamos el eventHandler. Recibe varios parámetros como son un objeto de tipo capture, un string que contiene el numero de serie del sensor y un objeto de tipo Sample. Adicionalmente implementa una interfaz correspondiente al eventHandler.

```
1 Referencias
Public Sub onComplete(Capture As Object, ReaderSerialNumber As String, Sample As Sample) Implements EventHandler.OnComplete
    Dim timestamp As Long = DateTime.Now.Ticks
    photonames = "salida" & timestamp & ".png"
    Dim rutaSalida As String = "C:/Users/5242999/Documents/queso/" & photonames

    Dim imagenbmp = ConvertirSampleMapaDeBits(Sample)
    imagenbmp.Save(rutaSalida, Imaging.ImageFormat.Png)
    MessageBox.Show("Id: " & photonames)
    mapas = ConvertirSampleMapaDeBits(Sample)

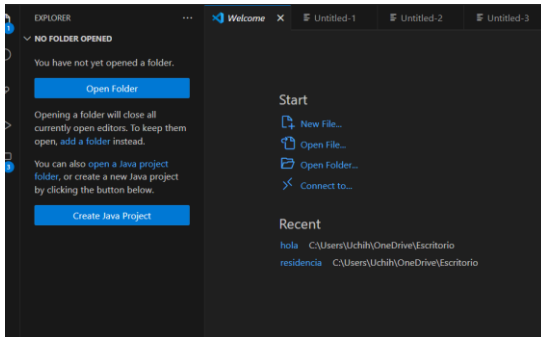
    ListPaths.Add(rutaSalida)
    MessageBox.Show(ListPaths.ToString)
```

Finalmente daremos clic en Compilar y al hacerlo nos generará un serie de carpetas y archivos, entre todos ellos estará el archivo .dll que ocuparemos. Es pues, la biblioteca que contiene todos los elementos que nos permitirá trabajar en Python.

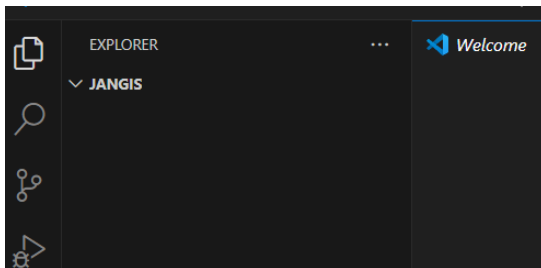
Compilar Depurar Prue...
Debug Any CPU
Class1

11.2.2 Entorno Virtual

El primer paso es abrir Visual Studio Code y seleccionar una carpeta para empezar el desarrollo del proyecto.

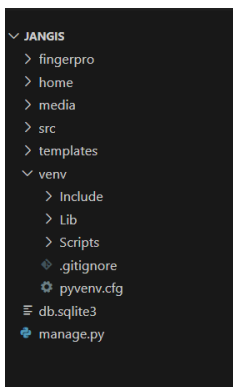


Cuando tengamos nuestra carpeta lo siguiente es crear un entorno virtual para poder utilizar django.

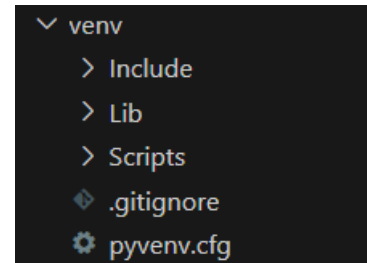


Para crear el entorno virtual lo haremos con la siguiente línea de código.

```
PS C:\Users\Uchi> python -m virtualenv -p python3 venv
```



Cuando se haya creado el entorno virtual se podrá ver de la siguiente manera:



El siguiente paso es instalar django. Para ello usaremos la siguiente línea de código.

```
(venv) C:\Users\r2d2999\Desktop\jangis>pip install django
Requirement already satisfied: django in c:\users\r2d2999\desktop\
Requirement already satisfied: asgiref<4,>=3.7.0 in c:\users\r2d2999\
Requirement already satisfied: sqlparse>=0.3.1 in c:\users\r2d2999\
Requirement already satisfied: tzdata in c:\users\r2d2999\desktop\
```

Se comprueba que django esta instalado exitosamente.

Una vez que hayamos instalado django podemos realizar una prueba de verificación que todo salió correctamente.

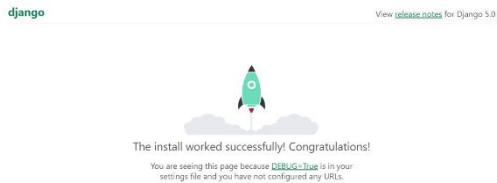
Lo haremos con la siguiente línea de código:

```
Python manage.py run server
```

Al hacerlo nos mostrará la siguiente pantalla:

```
You have 18 unapplied migration(s). Your project may not work properly.
Run 'python manage.py migrate' to apply them.
February 06, 2024 - 21:04:57
Django version 5.0.2, using settings 'fingerp.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Después daremos clic en esa dirección y nos abrirá el navegador web, donde, si todo salió bien, nos mostrará la siguiente pantalla:



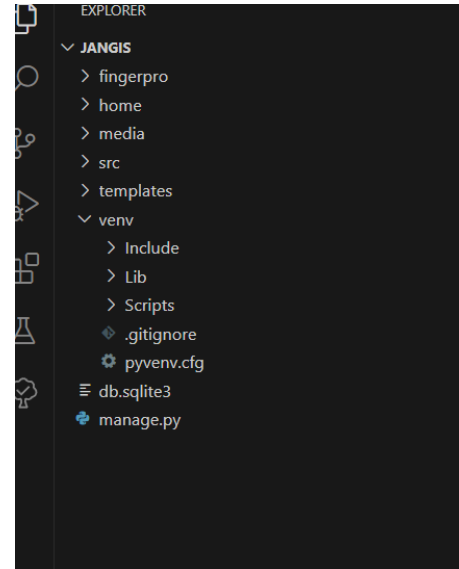
Lo siguiente será empezar con la programación del proyecto.

11.2.3 Programación en Django

La siguiente pantalla muestra el esquema de la estructura del proyecto. En el cual contamos con la aplicación home, la que es la que contiene los elementos propios del proyecto.

En un mismo proyecto podemos tener más de una aplicación, sin embargo. En

esta caso solamente estaremos trabajando con una sola.

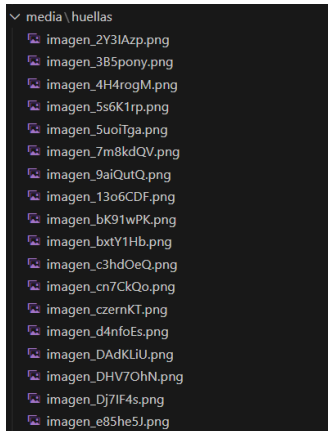


La aplicación contiene diversas carpetas que completan el funcionamiento del proyecto. Algunas carpetas tienen el mismo nombre que las carpetas del proyecto, aunque funcionan en un rango diferente. Similar al scope en programación orientada a objetos.

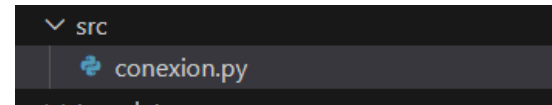
La carpeta media se utiliza para guardar los diversos archivos que componen los medios (imágenes, videos, etc) que se pueden utilizar en el proyecto.

Para este caso también está la subcarpeta huellas, que es donde se

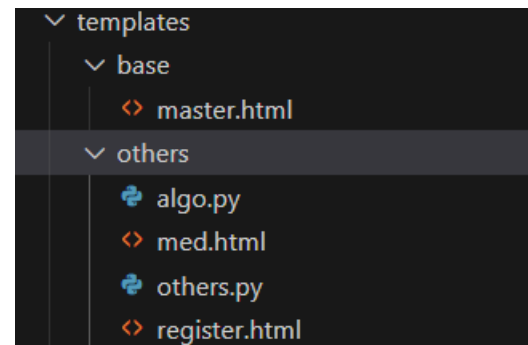
guardan las imágenes que el sensor de huellas digitales almacena.



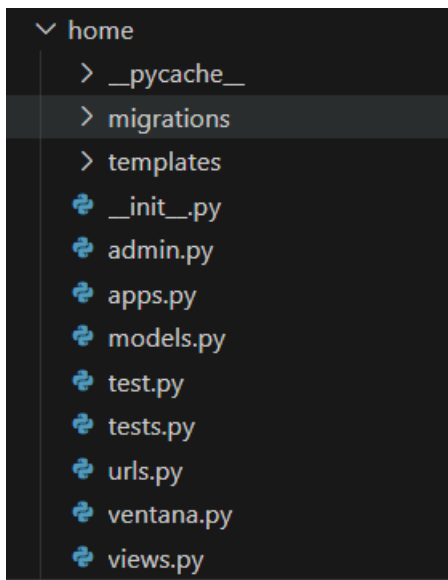
La cuál contiene el código necesario para realizar la conexión con la base de datos.



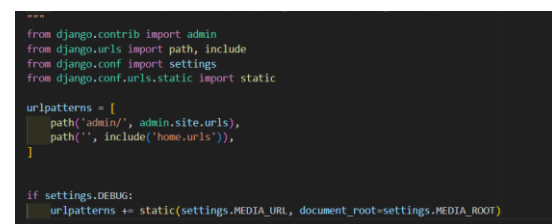
La carpeta other contiene algunos elementos que se han generado para realizar pruebas del desarrollo sin que pueda afectar fundamentalmente el proyecto mismo y su ejecución.



La carpeta migraciones se utiliza para almacenar las diferentes migraciones que hacemos en las bases de datos.



Dentro de la carpeta urls, del proyecto, encontramos el siguiente archivo. Este archivo se utiliza para registrar todas las diferentes urls que puedan existir dentro de las diversas aplicaciones que podamos lanzar.



La carpeta src la utilizamos para guardar la clase:

Conexion.py

Dentro de la estructura de la aplicación también encontraremos al archivo `settings.py`. El cuál nos permite realizar varias configuraciones del proyecto. En este caso se muestra el código correspondiente a las configuraciones de la base de datos. Por defecto utiliza `sqlite3`, para el proyecto se ha decidido usar `postgres`.

```
settings.py X
fingerpro > settings.py > ...
79
80 # settings.py
81
82 DATABASES = {
83     'default': {
84         'ENGINE': 'django.db.backends.postgres',
85         'NAME': 'personas',
86         'USER': 'postgres',
87         'PASSWORD': 'password',
88         'HOST': 'localhost',
89         'PORT': '5432',
90     }
91 }
92
93
94
95 # Password validation
```

El archivo `admin.py` se encarga de poder registrar los diferentes modelos con los que vamos a trabajar. Mismos que podrán ser visualizados y modificados dentro del apartado del administrador del proyecto en `django`.

Es necesario registrar todos los modelos con los que estemos trabajando.

```
admin.py X
home > admin.py
1 from django.contrib import admin
2 from .models import Task, Persona
3
4
5 # Register your models here.
6
7 admin.site.register(Task)
8 admin.site.register(Persona)
9
10
11
```

El siguiente fragmento de código corresponde a la clase `models.py`

```
models.py X
home > models.py > People register > @login
1 from django.db import models
2
3 # Create your models here.
4
5
6 class Task(models.Model):
7     description = models.CharField(max_length=255)
8     done = models.BooleanField(default=False)
9
10
11 class Persona(models.Model):
12     ids = models.AutoField(primary_key=True)
13     nombre = models.CharField(max_length=255)
14     apellido = models.CharField(max_length=255)
15     imagen = models.ImageField(upload_to='imagenes/', null=True, blank=True)
16
17
18 class Producto(models.Model):
19     nombre = models.CharField(max_length=100)
20     precio = models.DecimalField(max_digits=10, decimal_places=2)
21     descripcion = models.TextField()
22
23
24
25 class People_register(models.Model):
26     nombre = models.CharField(max_length=100)
```

Se puede ver como se han definido cuatro clases diferentes que corresponden a cuatro bases de datos diferentes y cada una con datos y atributos propios.

Dentro de la aplicación `home` también encontramos un archivo `urls.py`, siendo este propio y único de la aplicación. Dentro de esta clase definimos un diccionario de direcciones (`urls`) que componen la navegación de las páginas web de la aplicación.

```
urls.py
home > urls.py > ...
1 from django.urls import path
2 from .views import homeView, createTaskView, registerView, neoMainView
3
4 urlpatterns=[
5     path('', homeView, name='home'),
6     path('create_task', createTaskView, name='createTask'),
7     path('registerdata/', registerView, name='registerdata'),
8     path('main/', neoMainView, name='neo'),
9 ]
10 ]
```

El siguiente fragmento de código es de vital importancia, porque es donde hacemos la conexión directa entre el código de las páginas web y las plantillas.

```
views.py
home > views.py > neoMainView
100
101 def main_thread():
102
103
104
105
106 # create your views here.
107
108
109
110
111
112
113
114
115
116 def homeView(request):
117     avens = ventanaDOS()
118     avens = InterupExplorer()
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

El archivo views.py nos permite crear diferentes métodos para realizar todo aquello que necesitemos en el funcionamiento de las páginas web del proyecto. También permite renderizar una plantilla html a una función en específico.

Dicho de otra manera: asigna una plantilla de html a una función de Python y los diferentes métodos que esta pueda requerir.

En el archivo views.py vamos a utilizar una serie de diversos imports de varias bibliotecas que nos serán necesarias. Como se puede apreciar se utilizan bibliotecas propias de Python y algunas externas como lo son clr (biblioteca de pythonnet) y demás.

```
views.py
home > views.py > neoMainView
1 from django.shortcuts import render, redirect, reverse
2 from django.http import HttpResponseRedirect
3 from .models import Task, Persona, People_register
4 import clr
5 import tkinter as tk
6 from PIL import Image, ImageTk
7 from django.core.files import File
8 import json
9
10
11 clr.AddReference("C:/Users/r2d2999/Documents/py/DPFPShrNET.dll")
12 clr.AddReference("C:/Users/r2d2999/Documents/py/DPFPDevNET.dll")
13 clr.AddReference("C:/Users/r2d2999/Documents/py/DPFPEngNET.dll")
14 clr.AddReference("C:/Users/r2d2999/Desktop/s/camisa.dll")
15
16
17 from camisa import Class1
18 from DPFP import Capture
19 from DPFP.Processing import Enrollment
20 from DPFP import Template
21
22
23
24
```

Después de realizar los imports correspondientes ahora hacemos uso de la biblioteca de pythonnet (clr) para poder utilizar el método *AddReference* el cual nos permite, básicamente, agregar referencias. Tal cual haríamos en un proyecto de visual basic.

Añadir estas referencias es lo que hará que ocurra la magia; podremos utilizar los métodos propios y nativos del sdk de Digital Persona. Mismos que de otro modo, nunca habríamos podido acceder, al menos no, de manera nativa.

Un vez que agregamos las referencias necesarias para poder trabajar, haremos uso de los métodos del sdk de Digital Persona mandando a traerlos de la misma manera en que lo haríamos con bibliotecas de Python. Solo que

ahora el SDK es donde se tomara todo lo necesario para trabajar.

Al utilizar pythonnet dentro del entorno de desarrollo de Django tenemos poderosas herramientas para el desarrollo del proyecto, ya que al utilizar esta librería podemos acceder de una manera bastante fácil a los elementos gráficos de .net (visual Basic) logrando de esta manera que funcionen, casi como de manera nativa en el desarrollo de un proyecto utilizar Python como código fuente.

```
home > views.py > neoMainView
29
30 clr.AddReference('System.IO')
31 clr.AddReference('System.Drawing')
32 clr.AddReference('System.Reflection')
33 clr.AddReference('System.Threading')
34 clr.AddReference('System.Windows.Forms')
35
36 import System
37 import System.IO
38 import System.Drawing
39 import System.Threading
40 import System.Reflection
41 import System.Windows.Forms
42 import System.Windows.Forms
43
44 from System.Windows.Forms import Form
45 from System.Windows.Forms import MessageBox
46 from System.Windows.Forms import PictureBox, PictureBoxSizeMode
47 from System.Drawing import Image
48 from System.Windows.Forms import Label
49 from System.Windows.Forms import TextBox
50
51
52 from System.Threading import Thread, ThreadStart, ApartmentState
53
54
```

Añadimos referencias propias de .net para la implementación y el uso de todos los elementos gráficos .net que necesitaremos.

Posteriormente vamos a realizar los imports necesarios para acceder al contenido de las bibliotecas que contienen los elementos gráficos de .net.

Al momento de realizar las diversas funciones que necesita al proyecto, hemos de hacer un especial énfasis en el método *homeView*, mismo que tiene como atributo un objeto de clase *request* (se encarga de manejar las solicitudes entre plantillas HTML) y que regresa un objeto de clase *render* (necesario para poder mostrar la plantilla HTML en la que estamos ligando la función).

```
15
16 def homeView(request):
17     #venx = ventanaDOS()
18     #venx = InteropExplorer()
19
20
21
22 > if request.POST: ...
23
24
25
26
27
28
29
30
31     context = {
32         'message': 'tengo comeson',
33         'tasks': Task.objects.all()
34     }
35
36
37
38
39
40     return render(request, 'base/master.html', context)
41
42
```

Dentro del método *homeView* tenemos el siguiente código:

```
if request.POST:
    print("algo")
    #venx.mainloop()
    main_form_thread()

context = {
    'message': 'tengo comeson',
    'tasks': Task.objects.all()
}

return render(request, 'base/master.html', context)
```

Dentro de una condicional *if* hemos definido que si en el atributo *request* tenemos la llamada a un método POST entonces mandaremos a llamar al

método `main_form_thread` el cual se encarga de mostrar la pantalla y la ejecución correspondiente a la parte del sensor de huellas digitales. Es de suma importancia mencionar que este método nos permite emular el funcionamiento de un hilo y trabajar como si Python permitiera hacerlo y, a su vez, nos permite usar diversos hilos al mismo tiempo.

Podemos observar en el código del método `main_form_thread` que nada más empezar hacemos llamado a una instancia de otro método llamado `InteropExplorer`, mismo que es el método encargado de crear el diseño de la pantalla que contiene la interfaz grafica que hemos creado con ayuda de las bibliotecas de .net

```
390
391 def main_form_thread():
392
393     #Inicializar el form
394     interop_form = InteropExplorer()
395
396     #impresion en consola
397     print("se creo la instancia :D")
398
399     #definir el nuevo objeto de forms
400     win_form_app = System.Windows.Forms.Application
401
402     #Correr el interop explorer
403
404
405     ds.Init()
406     ds.IniciarCaptura()
407
408     win_form_app.Run(interop_form)
409
410
411
412     # Create your views here.
413
```

Después de que se crea la instancia se crea una variable `win_form_app` y la definimos como un objeto del tipo `forms` de .net.

Mandamos a atraer los métodos que corresponden al sdk de Digital Persona

y finalmente corremos la variable de tipo `forms`.

La clase `InteropExplorer` tiene como atributo un objeto de clase `forms` de .net. Dentro de la clase contamos con diversos métodos que, en conjunto y en principio, nos permiten emular casi perfectamente, el ciclo de vida de un objeto del tipo `forms`.

Muchos de los eventos a los que responde el objeto `Form` pertenecen al ciclo de vida del formulario

Entre estos eventos se encuentran los siguientes, en orden de ocurrencia:

- Load: El formulario está en memoria, pero invisible.
- Paint: Se "pinta" el formulario y sus controles.
- Activated: El formulario recibe foco.
- FormClosing: Permite cancelar el cierre.
- FormClosed: El formulario es invisible.
- Disposed: El objeto está siendo destruido.

Dado que no hay una manera sencilla ni mucho menos nativa de poder utilizar estos métodos y a su vez de intervenir en ellos o en lo que podemos hacer o no mientras ocurren estos sucesos, hemos de buscar la manera de simularlos para que el proceso pueda ocurrir lo más parecido al entorno original.

```
class InteropExplorer(Form):  
  
    > def __init__(self): ...  
  
    > def btnOnClick(self, sender, args): ...  
  
    > def btnOnClickClose(self, sender, args): ...  
  
    def dispose(self):  
        self.components.Dispose()  
        System.Windows.Forms.Form.Dispose(self)  
  
    def run(self):
```

Comenzamos definiendo un constructor donde creamos variables que vamos a inicializar con los diversos elementos gráficos (widgets) que utilizaremos en la creación de la pantalla de .net.

```
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000  
1001  
1002  
1003  
1004  
1005  
1006  
1007  
1008  
1009  
1010  
1011  
1012  
1013  
1014  
1015  
1016  
1017  
1018  
1019  
1020  
1021  
1022  
1023  
1024  
1025  
1026  
1027  
1028  
1029  
1030  
1031  
1032  
1033  
1034  
1035  
1036  
1037  
1038  
1039  
1040  
1041  
1042  
1043  
1044  
1045  
1046  
1047  
1048  
1049  
1050  
1051  
1052  
1053  
1054  
1055  
1056  
1057  
1058  
1059  
1060  
1061  
1062  
1063  
1064  
1065  
1066  
1067  
1068  
1069  
1070  
1071  
1072  
1073  
1074  
1075  
1076  
1077  
1078  
1079  
1080  
1081  
1082  
1083  
1084  
1085  
1086  
1087  
1088  
1089  
1090  
1091  
1092  
1093  
1094  
1095  
1096  
1097  
1098  
1099  
1100  
1101  
1102  
1103  
1104  
1105  
1106  
1107  
1108  
1109  
1110  
1111  
1112  
1113  
1114  
1115  
1116  
1117  
1118  
1119  
1120  
1121  
1122  
1123  
1124  
1125  
1126  
1127  
1128  
1129  
1130  
1131  
1132  
1133  
1134  
1135  
1136  
1137  
1138  
1139  
1140  
1141  
1142  
1143  
1144  
1145  
1146  
1147  
1148  
1149  
1150  
1151  
1152  
1153  
1154  
1155  
1156  
1157  
1158  
1159  
1160  
1161  
1162  
1163  
1164  
1165  
1166  
1167  
1168  
1169  
1170  
1171  
1172  
1173  
1174  
1175  
1176  
1177  
1178  
1179  
1180  
1181  
1182  
1183  
1184  
1185  
1186  
1187  
1188  
1189  
1190  
1191  
1192  
1193  
1194  
1195  
1196  
1197  
1198  
1199  
1200  
1201  
1202  
1203  
1204  
1205  
1206  
1207  
1208  
1209  
1210  
1211  
1212  
1213  
1214  
1215  
1216  
1217  
1218  
1219  
1220  
1221  
1222  
1223  
1224  
1225  
1226  
1227  
1228  
1229  
1230  
1231  
1232  
1233  
1234  
1235  
1236  
1237  
1238  
1239  
1240  
1241  
1242  
1243  
1244  
1245  
1246  
1247  
1248  
1249  
1250  
1251  
1252  
1253  
1254  
1255  
1256  
1257  
1258  
1259  
1260  
1261  
1262  
1263  
1264  
1265  
1266  
1267  
1268  
1269  
1270  
1271  
1272  
1273  
1274  
1275  
1276  
1277  
1278  
1279  
1280  
1281  
1282  
1283  
1284  
1285  
1286  
1287  
1288  
1289  
1290  
1291  
1292  
1293  
1294  
1295  
1296  
1297  
1298  
1299  
1300  
1301  
1302  
1303  
1304  
1305  
1306  
1307  
1308  
1309  
1310  
1311  
1312  
1313  
1314  
1315  
1316  
1317  
1318  
1319  
1320  
1321  
1322  
1323  
1324  
1325  
1326  
1327  
1328  
1329  
1330  
1331  
1332  
1333  
1334  
1335  
1336  
1337  
1338  
1339  
1340  
1341  
1342  
1343  
1344  
1345  
1346  
1347  
1348  
1349  
1350  
1351  
1352  
1353  
1354  
1355  
1356  
1357  
1358  
1359  
1360  
1361  
1362  
1363  
1364  
1365  
1366  
1367  
1368  
1369  
1370  
1371  
1372  
1373  
1374  
1375  
1376  
1377  
1378  
1379  
1380  
1381  
1382  
1383  
1384  
1385  
1386  
1387  
1388  
1389  
1390  
1391  
1392  
1393  
1394  
1395  
1396  
1397  
1398  
1399  
1400  
1401  
1402  
1403  
1404  
1405  
1406  
1407  
1408  
1409  
1410  
1411  
1412  
1413  
1414  
1415  
1416  
1417  
1418  
1419  
1420  
1421  
1422  
1423  
1424  
1425  
1426  
1427  
1428  
1429  
1430  
1431  
1432  
1433  
1434  
1435  
1436  
1437  
1438  
1439  
1440  
1441  
1442  
1443  
1444  
1445  
1446  
1447  
1448  
1449  
1450  
1451  
1452  
1453  
1454  
1455  
1456  
1457  
1458  
1459  
1460  
1461  
1462  
1463  
1464  
1465  
1466  
1467  
1468  
1469  
1470  
1471  
1472  
1473  
1474  
1475  
1476  
1477  
1478  
1479  
1480  
1481  
1482  
1483  
1484  
1485  
1486  
1487  
1488  
1489  
1490  
1491  
1492  
1493  
1494  
1495  
1496  
1497  
1498  
1499  
1500  
1501  
1502  
1503  
1504  
1505  
1506  
1507  
1508  
1509  
1510  
1511  
1512  
1513  
1514  
1515  
1516  
1517  
1518  
1519  
1520  
1521  
1522  
1523  
1524  
1525  
1526  
1527  
1528  
1529  
1530  
1531  
1532  
1533  
1534  
1535  
1536  
1537  
1538  
1539  
1540  
1541  
1542  
1543  
1544  
1545  
1546  
1547  
1548  
1549  
1550  
1551  
1552  
1553  
1554  
1555  
1556  
1557  
1558  
1559  
1560  
1561  
1562  
1563  
1564  
1565  
1566  
1567  
1568  
1569  
1570  
1571  
1572  
1573  
1574  
1575  
1576  
1577  
1578  
1579  
1580  
1581  
1582  
1583  
1584  
1585  
1586  
1587  
1588  
1589  
1590  
1591  
1592  
1593  
1594  
1595  
1596  
1597  
1598  
1599  
1600  
1601  
1602  
1603  
1604  
1605  
1606  
1607  
1608  
1609  
1610  
1611  
1612  
1613  
1614  
1615  
1616  
1617  
1618  
1619  
1620  
1621  
1622  
1623  
1624  
1625  
1626  
1627  
1628  
1629  
1630  
1631  
1632  
1633  
1634  
1635  
1636  
1637  
1638  
1639  
1640  
1641  
1642  
1643  
1644  
1645  
1646  
1647  
1648  
1649  
1650  
1651  
1652  
1653  
1654  
1655  
1656  
1657  
1658  
1659  
1660  
1661  
1662  
1663  
1664  
1665  
1666  
1667  
1668  
1669  
1670  
1671  
1672  
1673  
1674  
1675  
1676  
1677  
1678  
1679  
1680  
1681  
1682  
1683  
1684  
1685  
1686  
1687  
1688  
1689  
1690  
1691  
1692  
1693  
1694  
1695  
1696  
1697  
1698  
1699  
1700  
1701  
1702  
1703  
1704  
1705  
1706  
1707  
1708  
1709  
1710  
1711  
1712  
1713  
1714  
1715  
1716  
1717  
1718  
1719  
1720  
1721  
1722  
1723  
1724  
1725  
1726  
1727  
1728  
1729  
1730  
1731  
1732  
1733  
1734  
1735  
1736  
1737  
1738  
1739  
1740  
1741  
1742  
1743  
1744  
1745  
1746  
1747  
1748  
1749  
1750  
1751  
1752  
1753  
1754  
1755  
1756  
1757  
1758  
1759  
1760  
1761  
1762  
1763  
1764  
1765  
1766  
1767  
1768  
1769  
1770  
1771  
1772  
1773  
1774  
1775  
1776  
1777  
1778  
1779  
1780  
1781  
1782  
1783  
1784  
1785  
1786  
1787  
1788  
1789  
1790  
1791  
1792  
1793  
1794  
1795  
1796  
1797  
1798  
1799  
1800  
1801  
1802  
1803  
1804  
1805  
1806  
1807  
1808  
1809  
1810  
1811  
1812  
1813  
1814  
1815  
1816  
1817  
1818  
1819  
1820  
1821  
1822  
1823  
1824  
1825  
1826  
1827  
1828  
1829  
1830  
1831  
1832  
1833  
1834  
1835  
1836  
1837  
1838  
1839  
1840  
1841  
1842  
1843  
1844  
1845  
1846  
1847  
1848  
1849  
1850  
1851  
1852  
1853  
1854  
1855  
1856  
1857  
1858  
1859  
1860  
1861  
1862  
1863  
1864  
1865  
1866  
1867  
1868  
1869  
1870  
1871  
1872  
1873  
1874  
1875  
1876  
1877  
1878  
1879  
1880  
1881  
1882  
1883  
1884  
1885  
1886  
1887  
1888  
1889  
1890  
1891  
1892  
1893  
1894  
1895  
1896  
1897  
1898  
1899  
1900  
1901  
1902  
1903  
1904  
1905  
1906  
1907  
1908  
1909  
1910  
1911  
1912  
1913  
1914  
1915  
1916  
1917  
1918  
1919  
1920  
1921  
1922  
1923  
1924  
1925  
1926  
1927  
1928  
1929  
1930  
1931  
1932  
1933  
1934  
1935  
1936  
1937  
1938  
1939  
1940  
1941  
1942  
1943  
1944  
1945  
1946  
1947  
1948  
1949  
1950  
1951  
1952  
1953  
1954  
1955  
1956  
1957  
1958  
1959  
1960  
1961  
1962  
1963  
1964  
1965  
1966  
1967  
1968  
1969  
1970  
1971  
1972  
1973  
1974  
1975  
1976  
1977  
1978  
1979  
1980  
1981  
1982  
1983  
1984  
1985  
1986  
1987  
1988  
1989  
1990  
1991  
1992  
1993  
1994  
1995  
1996  
1997  
1998  
1999  
2000  
2001  
2002  
2003  
2004  
2005  
2006  
2007  
2008  
2009  
2010  
2011  
2012  
2013  
2014  
2015  
2016  
2017  
2018  
2019  
2020  
2021  
2022  
2023  
2024  
2025  
2026  
2027  
2028  
2029  
2030  
2031  
2032  
2033  
2034  
2035  
2036  
2037  
2038  
2039  
2040  
2041  
2042  
2043  
2044  
2045  
2046  
2047  
2048  
2049  
2050  
2051  
2052  
2053  
2054  
2055  
2056  
2057  
2058  
2059  
2060  
2061  
2062  
2063  
2064  
2065  
2066  
2067  
2068  
2069  
2070  
2071  
2072  
2073  
2074  
2075  
2076  
2077  
2078  
2079  
2080  
2081  
2082  
2083  
2084  
2085  
2086  
2087  
2088  
2089  
2090  
2091  
2092  
2093  
2094  
2095  
2096  
2097  
2098  
2099  
2100  
2101  
2102  
2103  
2104  
2105  
2106  
2107  
2108  
2109  
2110  
2111  
2112  
2113  
2114  
2115  
2116  
2117  
2118  
2119  
2120  
2121  
2122  
2123  
2124  
2125  
2126  
2127  
2128  
2129  
2130  
2131  
2132  
2133  
2134  
2135  
2136  
2137  
2138  
2139  
2140  
2141  
2142  
2143  
2144  
2145  
2146  
2147  
2148  
2149  
2150  
2151  
2152  
2153  
2154  
2155  
2156  
2157  
2158  
2159  
2160  
2161  
2162  
2163  
2164  
2165  
2166  
2167  
2168  
2169  
2170  
2171  
2172  
2173  
2174  
2175  
2176  
2177  
2178  
2179  
2180  
2181  
2182  
2183  
2184  
2185  
2186  
2187  
2188  
2189  
2190  
2191  
2192  
2193  
219
```

11.2.3 Imágenes

El siguiente método permite mostrar las imágenes que el sensor captura, en elemento de .net llamado picture box.

```
campo = self.ent.get()
campo2 = self.ent2.get()
campo = list()
print(campo)
#verifica campos vacios
if self.checkDataOn(campo, campo2):
    print("Trueeeee")
    ds.Init()
    ds.IniciarCaptura()
    #self.saveData(campo, campo2)
else:
    print("falseeeee")
def on_click_print(self):
    print("imagen show")
    self.mostrar_imagen()
```

Adicionalmente, se crea una variable que va a ser igual a una instancia del modelo Persona, el cual contiene datos que se van a guardar en la base de datos, al igual que el nombre de la imagen.

```
instancia = Persona(nombre = "Juan", apellido = "Quecos")
with open(ruta_save, 'rb') as archivo:
    image_file = File(archivo)
    instancia.imagen.save('imagen.png', image_file)
instancia.save()
```

El siguiente método nos permite guardar las imágenes en la base de datos.

Para ello, lo primero que hacemos es recuperar el nombre del archivo por medio de la variable photonames, accedemos a ella por medio de el objeto ds, que pertenece a una variable original del archivo .dll de la biblioteca que estamos utilizando de Digital persona.

A continuación, vamos a guardar el registro en la base de datos y para ello, lo primero que hacemos es una validación para que los campos de nombre y apellido no estén vacíos. Si es el caso podemos proseguir.

Posteriormente, con ayuda de un try - catch, realizamos el proceso de guardado. De ser exitoso el proceso nos mostrará un mensaje y de lo contrario nos mostrará el error del porqué no se pudo guardar.

```
print(ape)
ruta_saveimg = "C:/Users/r2d2999/Documents/queso/" + ds.photonames

if name and ape:
    try:
        person = People_register(nombre = name, apellido = ape)
        person.save()

        with open(ruta_saveimg, 'rb') as archivo:
            image_file = File(archivo)
            person.imagen.save('imagen.png', image_file)
            person.save()

        #print("lista: " + ds.listPaths)
        MessageBox.Show("guardado con éxito")

        self.txtb.Text = ""
        self.txtb2.Text = ""

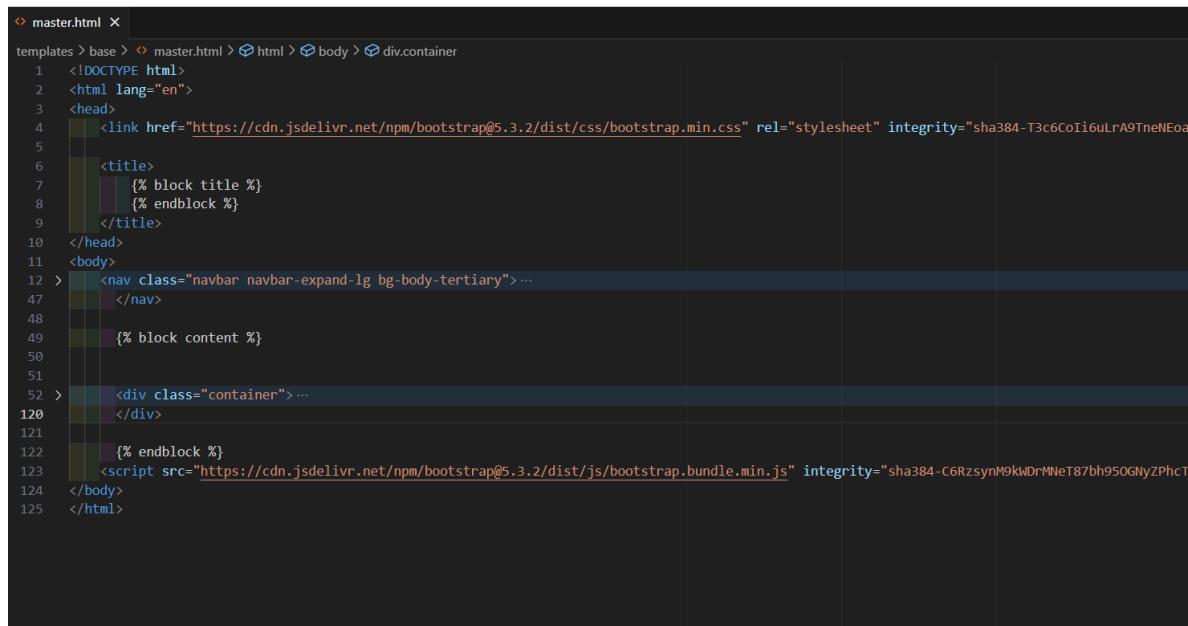
    except Exception as e:
        MessageBox.Show("Error al guardar")
    else:
        MessageBox.Show("No hay datos")
```

11.2.4 Plantillas

El siguiente archivo HTML corresponde a la plantilla máster, de la carpeta plantillas/base de nuestra aplicación Home.

Recibe el nombre de master.html porque es la plantilla principal, de la que podemos, heredar, por así decirlo, la mayoría de las componentes hacia otras múltiples plantillas y no sea necesaria crear la estructura desde cero de todas las demás plantillas.

Comprobamos que los elementos de un HTML tradicional existen, la etiqueta head, body y HTML. Sin embargo, hacemos uso también de etiquetas propias de Django que nos permiten interactuar con las plantillas HTML de una manera más simple y eficaz. Tal es el caso de los Bloques.



```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-T3c6CoIi6uLrA9TneEOa
5
6   <title>
7     {% block title %}
8     {% endblock %}
9   </title>
10 </head>
11 <body>
12   <nav class="navbar navbar-expand-lg bg-body-tertiary">...
47   </nav>
48
49   {% block content %}
50
51
52   <div class="container">...
120  </div>
121
122   {% endblock %}
123   <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-C6RzsynM9kWDmWet87bh95OGHyZPhcT
124 </body>
125 </html>
```

Podemos ver que Utilizamos dos bloques:

- `{% block title %}`
- `{% block content %}`

Cada bloque se encarga de poder encapsular ciertos elementos van a ser heredados a otras plantillas que hereden la plantilla máster. Para el caso de *block title*, se encargará de guardar el título de la página y que no será cambiado en ninguna otra (salvo sea necesario), mientras que el *block content* contiene los elementos que van dentro de la etiqueta body del HTML.

Mismos que se repetirán según se requiera.

El siguiente código HTML corresponde a una etiqueta del tipo nav, misma que utiliza la clase *navbar* de Bootstrap. La cual nos permite hacer uso de elementos css ya predefinidos y modificables por medio del uso de las clases. También nos permite hacer uso de métodos de JavaScript.

La *navbar* contiene dentro una etiqueta de tipo div y adquiere la clase *container-fluid* de Bootstrap. Adicionalmente contiene una etiqueta botón, con la clase *navbar-toggler* que dará el aspecto que esta buscando cuando la barra de navegación este oculta o se este mostrando.

Los elementos del menú de la barra de navegación están dentro de una etiqueta ul con la clase *navbar-nav* me de Bosstrap y cada elemento del menú pertenece a una etiqueta del tipo li con la clase *nav-item* de Bootstrap.

```
<nav class="navbar navbar-expand-lg bg-body-tertiary">
  <div class="container-fluid">
    <a class="navbar-brand" href="#">Navbar</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupp
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarSupportedContent">
      <ul class="navbar-nav me-auto mb-2 mb-lg-0">
        <li class="nav-item">
          <a class="nav-link active" aria-current="page" href="{% url 'neo' %}">Home</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#">Link</a>
        </li>
        <li class="nav-item dropdown">
          <a class="nav-link dropdown-toggle" href="#" role="button" data-bs-toggle="dropdown" aria-ex
            Dropdown
          </a>
          <ul class="dropdown-menu">
            <li><a class="dropdown-item" href="#">Action</a></li>
            <li><a class="dropdown-item" href="#">Another action</a></li>
            <li><hr class="dropdown-divider"></li>
            <li><a class="dropdown-item" href="#">Something else here</a></li>
          </ul>
        </li>
        <li class="nav-item">
          <a class="nav-link disabled" aria-disabled="true">Disabled</a>
        </li>
      </ul>
```

El siguiente código html corresponde al block content y podemos ver que todo ocurre dentro de una etiqueta de tipo div, que nos permite crear diversas divisiones en todo lo

ancho de la página web. Se le ha agregado la clase `container`, debido a que vamos a trabajar en un entorno responsivo.

Esto quiere decir que es adaptativo según los requerimientos de la pantalla donde se pudiera estar visualizando.

Aquí se comienza a generar la estructura HTML de la página con el uso de etiquetas `h1` para agregar texto, etiquetas `div` con la clase `row` de Bootstrap para generar una especie de estructura en malla para la disposición de cada elemento gráfico.

```
52 <div class="container">
53
54 <div class="row text-center">
55   <h1>Biometrica</h1>
56 </div>
57
58 <div class="row text-center">
59   <h1>Detector de Huellas Digitales</h1>
60 </div>
61
62 <div class="row mt-3">
63   <h5>Llena los campos si deseas registrarte o bien, has click en consultar si ya eres usuari
64 </div>
65
66 <div class="row mt-5">
67
68   <form method="POST">{% csrf_token %}
69
70     <div class="row">
71
72       <div class="col-lg-4">
73         <p>
74           Nombre(s) <input type="text" name="nombre">
75         </p>
76       </div>
77     </div>
78
79     <div class="col-lg-4">
80       <p>
81         Apellido(s) <input type="text" name="apellido">
82       </p>
83     </div>
84   </div>
85 </form>
86 </div>
```

Continuamos hasta llegar a la etiqueta `form` cuyo método es del tipo `POST` y dentro de esta existe otra etiqueta del tipo `div` que nos servirá para crear otra malla y orientar los elementos gráficos según sea a conveniencia.

12. Resultados Obtenidos

Dentro de visual Studio Code, en la terminal, escribiremos el siguiente comando, el cual nos permitirá abrir el proyecto.

```
You have 18 unapplied migration(s). Your project may not work properly.  
Run 'python manage.py migrate' to apply them.  
February 06, 2024 - 21:04:57  
Django version 5.0.2, using settings 'fingerp.settings'  
Starting development server at http://127.0.0.1:8000/  
Quit the server with CTRL-BREAK.  
1
```

Se procede a copiar la dirección del servidor y se pega en la barra de navegación de algún navegador, realmente no importa mucho cual. Cuando haya cargado la dirección, se mostrará la siguiente pantalla. Siendo esto la pantalla principal de la página web.

Se da clic en Registrar huella.

Navbar

Home

Link

Dropdown

Disabled

Search

Search

Biometrica

Detector de Huellas Digitales

Llena los campos si deseas registrarte o bien, has click en consultar si ya eres usuario

Nombre(s)

Apellido(s)

ID

Guardar

Iniciar Sesión

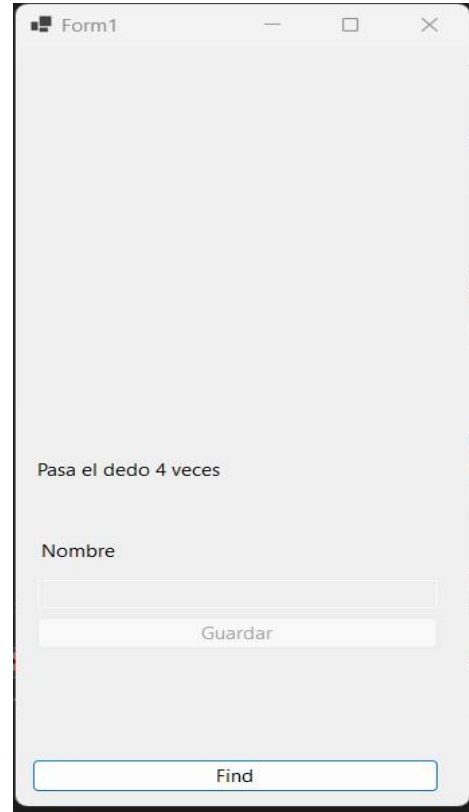
Registrar Huella

Al dar clic en registrar huella se mostrará la siguiente pantalla. Misma que corresponde a un elemento form de .net.

La siguiente pantalla nos pide agregar un nombre y una huella, para poder registrarlo y ejecutar los métodos que corresponden al guardado de los datos dentro de la base de datos.

Después de presionar una vez el dedo encima del lector del sensor de huellas digitales, el mensaje del elemento label cambiara, ahora nos pide que se presione 3 veces.

Continuaremos hasta que se complete el número de veces faltantes. Este valor viene por defecto en el SDK.



Form1

Pasa el dedo 4 veces

Nombre

Guardar

Find



Form1



Pasa el dedo 3 veces

Nombre

Guardar

Find



Form1



Pasa el dedo 2 veces

Nombre

Guardar

Find

Form1



Pasa el dedo 1 veces

Nombre

Guardar

Find

Form1



Pasa el dedo 0 veces

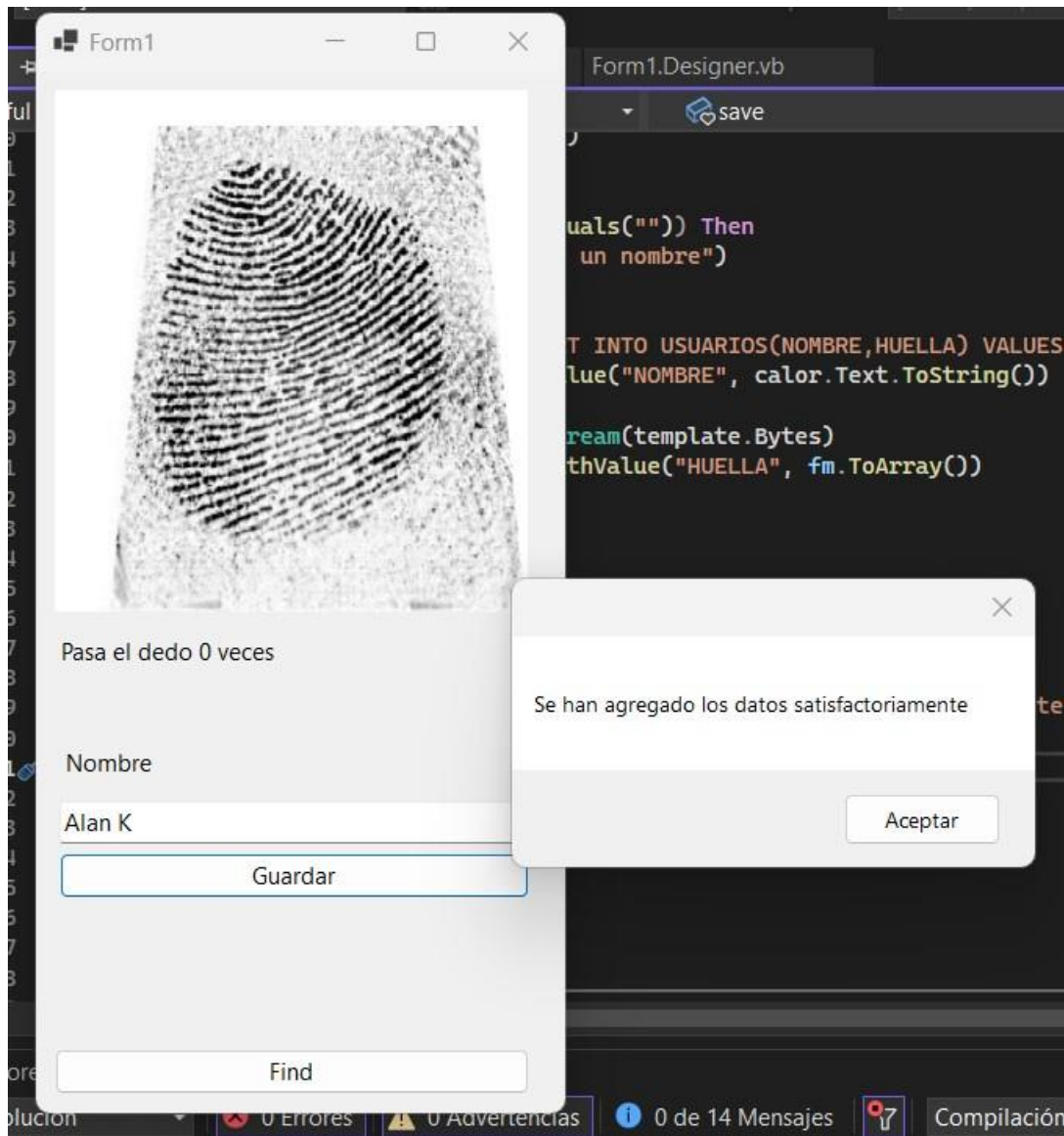
Nombre

Alan K

Guardar

Find

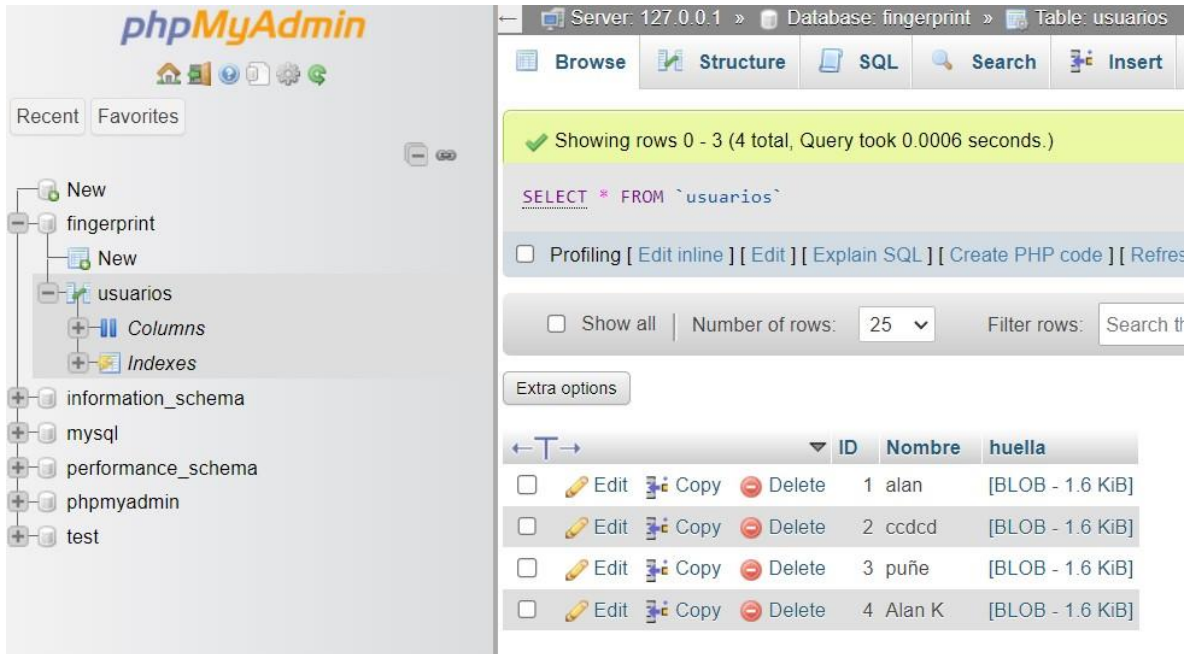
Después de que la cuenta regresiva termina y el sensor logró recuperar los datos que necesitaba para poder guardar las huellas y procesarlas, nos permite agregar un nombre y posteriormente, guardar el registro dentro de la base de datos.



Se comprueba que el registro se ha guardado satisfactoriamente al guardar el nombre de la persona y su huella.

Estos datos son enviados a la base de datos, donde podemos acceder a ellos y realizar diferentes funciones según se requieran.

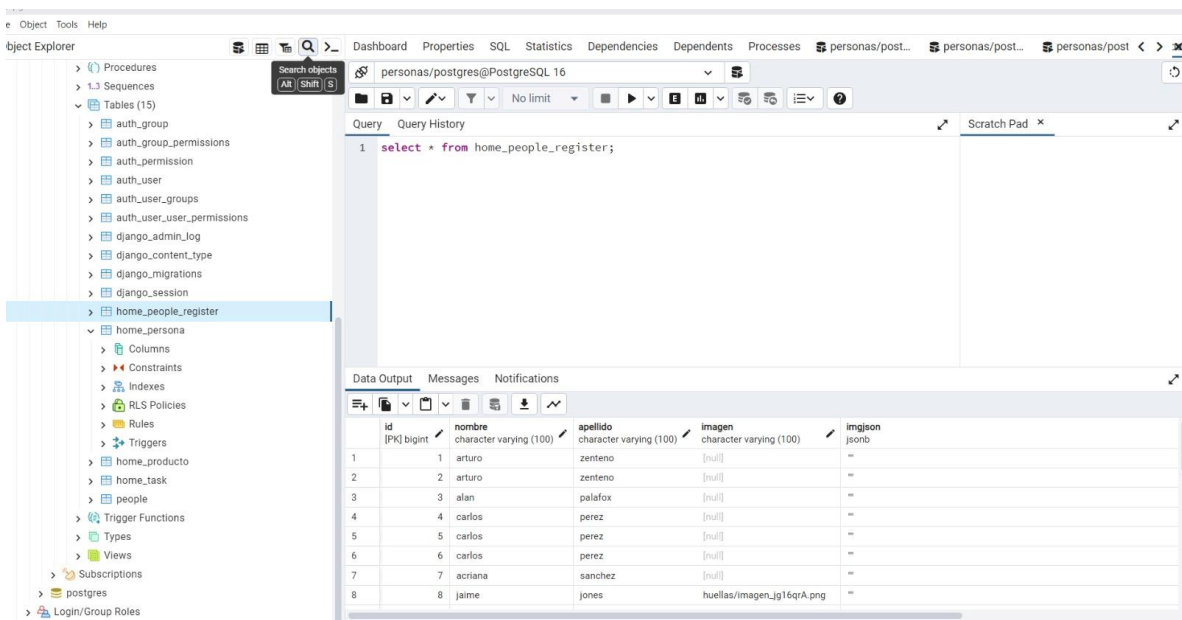
Accedemos a phpMyAdmin para corroborar que los registros se han estado guardando correctamente. En la parte lateral izquierda se puede ver el nombre de la base de datos “fingerprint” y el contenido de la tabla “usuarios”.



The screenshot shows the phpMyAdmin interface. On the left, the 'fingerprint' database is selected, and the 'usuarios' table is highlighted. The main panel shows the table structure and a list of records. The SQL query 'SELECT * FROM `usuarios`' is entered in the query box. The table has columns: ID, Nombre, and huella. The records are as follows:

ID	Nombre	huella
1	alan	[BLOB - 1.6 KiB]
2	ccdc	[BLOB - 1.6 KiB]
3	puñe	[BLOB - 1.6 KiB]
4	Alan K	[BLOB - 1.6 KiB]

Pantalla de Postgree SQL donde también se comprueba el guardado de los registros.

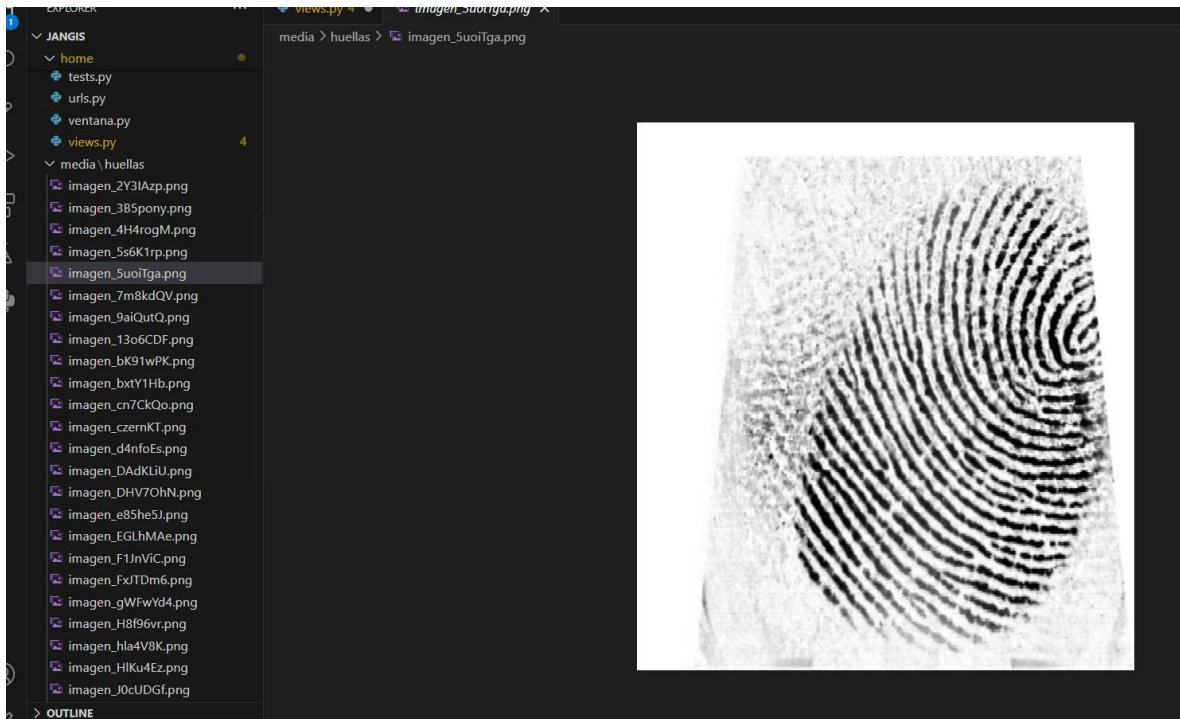


The screenshot shows the PostgreSQL SQL interface. The 'home_people_register' table is selected in the Object Explorer. The SQL query 'select * from home_people_register;' is entered in the query box. The table has columns: id, nombre, apellido, imagen, and imgjson. The records are as follows:

id	nombre	apellido	imagen	imgjson
1	arturo	zenteno	[null]	""
2	arturo	zenteno	[null]	""
3	alan	palafox	[null]	""
4	carlos	perez	[null]	""
5	carlos	perez	[null]	""
6	carlos	perez	[null]	""
7	acriana	sanchez	[null]	""
8	jaime	jones	huellas/imagen_jg16qrA.png	""

Carpeta que contiene las imágenes que se recuperan luego de la conversión de objetos de tipo Sample a .png.

El proceso ocurre de manera automática en el método sobre escrito OnComplete().



13. Conclusiones

Con el proyecto de análisis y experimentación para detección y reconocimiento de huellas dactilares se ha logrado diseñar e implementar un sistema de detección y reconocimiento de huellas dactilares que muestra un alto grado de precisión y eficiencia en la identificación de individuos.

Los algoritmos y técnicas de procesamiento de imágenes desarrollados han demostrado ser efectivos para mejorar la calidad de las huellas dactilares y reducir el impacto de ruidos o distorsiones en las imágenes capturadas.

El sistema ha superado los desafíos inherentes a la variabilidad de las huellas dactilares y ha mostrado una capacidad robusta para manejar diferentes escenarios y condiciones de uso.

El proyecto ha contribuido al avance en el campo de la biometría y el reconocimiento biométrico, al desarrollar y aplicar algoritmos avanzados de extracción y comparación de características únicas de las huellas dactilares.

Se han explorado y validado técnicas de aprendizaje automático y análisis de patrones que pueden ser aplicadas en otros contextos biométricos y de seguridad.

El proyecto ha contribuido al avance en el campo de la biometría y el reconocimiento biométrico, al desarrollar y aplicar algoritmos avanzados de extracción y comparación de características únicas de las huellas dactilares.

Se han explorado y validado técnicas de aprendizaje automático y análisis de patrones que pueden ser aplicadas en otros contextos biométricos y de seguridad.

El sistema desarrollado tiene aplicaciones prácticas en diversos sectores, como la seguridad informática, el control de acceso y la identificación personal en entornos gubernamentales, corporativos y comerciales.

Las mejoras en la precisión y eficiencia del reconocimiento de huellas dactilares pueden contribuir significativamente a la protección de datos personales y la prevención de fraudes y accesos no autorizados.

14. Competencias Desarrolladas

- Se adquieren conocimientos profundos sobre los principios y técnicas de la biometría aplicada a la identificación biométrica a través de huellas dactilares.
- Se desarrollan habilidades en el procesamiento y análisis de imágenes digitales para mejorar la calidad y extraer características relevantes de las huellas dactilares.
- Se obtiene experiencia en el desarrollo de algoritmos y software específicos para la detección y reconocimiento de huellas dactilares, utilizando lenguajes como Python, C++, entre otros.
- Se aprende a diseñar y llevar a cabo pruebas rigurosas para validar y evaluar el desempeño de sistemas biométricos, utilizando métricas de precisión, eficiencia y robustez.
- Se desarrollan habilidades en la aplicación de metodologías de investigación científica, incluyendo la revisión bibliográfica, la formulación de hipótesis, el diseño experimental y el análisis de resultados.
- Se adquieren capacidades para diseñar y ejecutar experimentos controlados que permitan validar y comparar diferentes enfoques y técnicas de detección y reconocimiento de huellas dactilares.
- Se aprende a utilizar herramientas y técnicas de análisis estadístico para interpretar los resultados obtenidos de los experimentos y sacar conclusiones válidas.
- Se mejora la capacidad de comunicar de manera efectiva los resultados, conclusiones y recomendaciones del proyecto tanto en informes escritos como en presentaciones orales.
- Se desarrollan habilidades para trabajar en equipo, colaborar con otros investigadores, supervisores y expertos en el campo, y coordinar actividades y tareas relacionadas con el proyecto.
- Se adquieren competencias en la planificación, gestión del tiempo y organización de tareas para cumplir con los plazos y objetivos del proyecto de tesis.

15. Referencias

1. Argentina, pionera de la dactiloscopia, BBC, 10 de noviembre de 2009
2. ↑ «Huellas Digitales | Evidencia Forense | Abogado Defensor». *criminallawyermaryland.net*. Consultado el 28 de marzo de 2016.
3. ↑ Texto completo del artículo de la investigación original (en inglés).
4. Andrew Orlowski and Carey Bishop. «MS Anti-Spyware built on MS Abandonware». The Register.
5. ↑ <http://www.microsoft.com/downloads/details.aspx?FamilyId=D7E31492-2595-49E6-8C02-1426FEC693AC&displaylang=en>
6. ↑ «VBOpenGL type library». Archivado desde el original el 16 de marzo de 2010.
7. ↑ Alex Homer, Dave Sussman, Rob Howard, Brian Francis, Karli Watson, Richard Anderson (2004). *Professional ASP.NET 1.1*. Wiley. ISBN 0764558900.
8. «Understanding the Import Address Table» (en inglés).
9. PEP 20 -- The Zen of Python
10. ↑ «Copia archivada». Archivado desde el original el 4 de agosto de 2018. Consultado el 25 de febrero de 2010.
11. ↑ Python Software Foundation. «More control flow options». *Python v2.7.8 Documentation* (en inglés). Consultado el 20 de julio de 2014.
12. ↑ Eric Huss. «Function Definition». *The C Library Reference Guide* (en inglés). Archivado desde el original el 18 de enero de 2015. Consultado el 20 de julio de 2014.
13. ↑ Álvarez, Miguel Ángel (2 de noviembre de 2001). «Funciones en Javascript». *desarrolloweb.com* (en inglés). Consultado el 20 de julio de 2014.
14. <http://www.microsoft.com/downloads/details.aspx?FamilyId=D7E31492-2595-49E6-8C02-1426FEC693AC&displaylang=en>
15. ↑ «VBOpenGL type library». Archivado desde el original el 16 de marzo de 2010.
16. ↑ Alex Homer, Dave Sussman, Rob Howard, Brian Francis, Karli Watson, Richard Anderson (2004). *Professional ASP.NET 1.1*. Wiley. ISBN 0764558900.
17. ↑ Marc D'Aoust. «Avoid Writing Tedious, Boring Code». Microsoft.
18. ↑ Andrew Troelsen (2008). *Pro VB 2008 and the .NET 3.5 Platform: The expert's voice in .NET*. Apress. p. 5. ISBN 1590598229.
19. Cuevas1208. (24 de Enero de 2020). github.com. Obtenido de github.com: https://github.com/cuevas1208/fingerprint_recognition
20. phddirection. (07 de Febrero de 2024). phddirection.com. Obtenido de phddirection.com: <https://www.phddirection.com/fingerprint-recognition-opencv-python/#:~:text=Fingerprint%20recognition%20is%20a%20method,in%20fingerprint%20valleys%20and%20ridges>