



TECNOLÓGICO NACIONAL DE MÉXICO

INSTITUTO TECNOLÓGICO DE APIZACO

Análisis y Experimentación para Extracción y Selección de Características para Huellas Dactilares

TITULACIÓN INTEGRAL INFORME DE RESIDENCIA PROFESIONAL

**PARA OBTENER EL TÍTULO PROFESIONAL DE:
INGENIERO EN TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIONES**

PRESENTA

ALAN HERNANDEZ PALAFOX

NOMBRE DEL ASESOR

M.I.A. Carlos Pérez Corona



APIZACO, TLAX, NOVIEMBRE 2024



Índice

1. Agradecimientos	3
2. Resumen	4
3. Introducción	5
4. Generalidades de la empresa	6
4.1. Mapa de la empresa	6
4.2. Logo de la empresa	7
4.3. Organigrama de la empresa	7
4.4. Acerca de la empresa	8
4.5. Misión y visión	8
4.6. Valores	8
5. Delimitación del problema	9
6. Objetivos	10
6.1. Objetivo general	10
6.2. Objetivos específicos	10
7. Justificación	11
8. Organigrama	12
9. Marco teórico	14
9.1. Huellas dactilares	14
9.2. Los fundamentos de la identificación de huellas dactilares	15
9.3. Funciones globales	16
9.4. Patrones básicos de crestas	17
9.5. Puntos Minuciosos	19
9.6. ¿Son las huellas dactilares realmente distintivas y persistentes?	20
10. Herramientas Tecnológicas Usadas	22
10.1. Lector biométrico U.are.U 4500 Reader de DigitalPersona	22
10.2. VisualBasic	23
10.3. Bibliotecas DLL	25
10.4. Python	26
10.5. Python.NET	27
10.6. VisualStudio 2022	28
10.7. VisualStudio Code	29
10.8. DotPeek	30
10.9. SDK DigitalPersona	32
10.10. Django	34
11. Desarrollo del proyecto	36
11.1. Instalación del SDK de DigitalPersona para sensor biométrico	36
11.2. Instalación de VisualStudio	38
11.3. Creación de la Biblioteca DLL	40
11.4. Instalación de dotPeek	45
11.5. Instalación de VisualStudioCode	50
11.6. Instalación de Python	57
11.7. Creación del Proyecto Django	61
11.8. Código Django	64
12. Resultados obtenidos	79
13. Conclusiones	88
14. Competencias desarrolladas	89
15. Referencias	90

1. Agradecimientos

Quiero expresar mi más sincero agradecimiento a la directora del Instituto Tecnológico de Apizaco: Yesica Imelda Saavedra Benítez por brindarme la invaluable oportunidad de realizar mis estudios en esta prestigiosa institución. Desde el primer día, me he sentido acogido/a por la calidez y el profesionalismo que caracterizan al Instituto Tecnológico de Apizaco.

La visión y el liderazgo de la directora Saavedra Benítez han sido fundamentales para el desarrollo académico y personal de todos los estudiantes. Su dedicación y compromiso con la excelencia educativa han inspirado a toda la comunidad universitaria a alcanzar sus metas más ambiciosas.

Agradezco también al jefe del departamento de Sistemas y Computación: Alan Augusto Gallegos Cuéllar, quién ha sido el asesor externo de este proyecto y al profesor Carlos Pérez Corona, quien fue el asesor interno, y también a cada miembro del personal docente y administrativo por su constante apoyo y orientación a lo largo de mi trayectoria académica. Sus conocimientos y experiencia han enriquecido mi aprendizaje y han contribuido significativamente a mi formación integral.

Por último, quiero reconocer el espíritu de colaboración y camaradería que se vive en el Tecnológico de Apizaco. El ambiente de respeto y compañerismo ha sido fundamental para mi desarrollo personal y profesional, y estoy profundamente agradecido/a por haber tenido la oportunidad de formar parte de esta gran familia universitaria.

2. Resumen

El proyecto se centra en el desarrollo de un sistema de análisis y experimentación para la extracción y selección de características en huellas dactilares, utilizando principalmente lenguajes de programación como Python y Visual Basic. La combinación de estos dos entornos permitió aprovechar las fortalezas y características específicas de cada uno para lograr un rendimiento óptimo. La finalidad principal es mejorar los procesos de identificación biométrica mediante la optimización de las características extraídas de las huellas dactilares.

El proyecto se enfocó en la experimentación exhaustiva, evaluando diferentes enfoques para la extracción y selección de características con el objetivo de mejorar la precisión y eficacia del reconocimiento de huellas dactilares. El análisis de resultados se realizó mediante métricas específicas, y se llevaron a cabo ajustes continuos en el código para optimizar el rendimiento del sistema.

Para llevar a cabo este proyecto, se emplearon diversas herramientas y software, entre las que destacan Visual Studio, Visual Studio Code y Python. Estas plataformas proporcionaron un entorno de desarrollo robusto y flexible para la implementación de algoritmos de extracción y selección de características, aprovechando las herramientas avanzadas de desarrollo y la integración de interfaces gráficas proporcionadas por estos entornos. Python se utilizó para la programación de algoritmos específicos relacionados con la extracción y selección de características en huellas dactilares, aprovechando su flexibilidad y potencia en el ámbito del procesamiento de imágenes.

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 identificación biométrica, en particular el reconocimiento de huellas dactilares, ha emergido como una herramienta crucial en numerosos campos, desde la seguridad hasta la gestión de identidades. La singularidad y estabilidad de las huellas dactilares las convierten en un medio confiable para la autenticación de individuos. Sin embargo, para lograr un reconocimiento preciso y eficiente, es fundamental realizar un análisis exhaustivo y una selección adecuada de características.

Este proyecto se enfoca en el análisis y experimentación para la extracción y selección de características en huellas dactilares, con el propósito de mejorar la precisión y confiabilidad de los sistemas de identificación biométrica basados en este método. A través de la combinación de técnicas avanzadas de procesamiento de imágenes y algoritmos específicos, buscamos desarrollar una solución que supere los desafíos actuales en la identificación biométrica, como la variabilidad de las huellas dactilares y el ruido en las imágenes capturadas.

En este contexto, exploraremos diferentes enfoques y metodologías para la extracción y selección de características, evaluando su efectividad a través de experimentos rigurosos y análisis comparativos. Además, aprovecharemos herramientas y tecnologías de vanguardia, incluyendo lenguajes de programación como Python y entornos de desarrollo como Visual Studio, para implementar y optimizar nuestro sistema.

El resultado de este proyecto no solo contribuirá al avance en la tecnología de reconocimiento de huellas dactilares, sino que también tendrá aplicaciones prácticas en una amplia gama de campos, desde la seguridad en aeropuertos y fronteras hasta la gestión de acceso en instituciones gubernamentales y corporativas. En última instancia, nuestro objetivo es desarrollar una solución robusta y confiable que mejore significativamente la precisión y eficiencia de los sistemas de identificación biométrica 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.

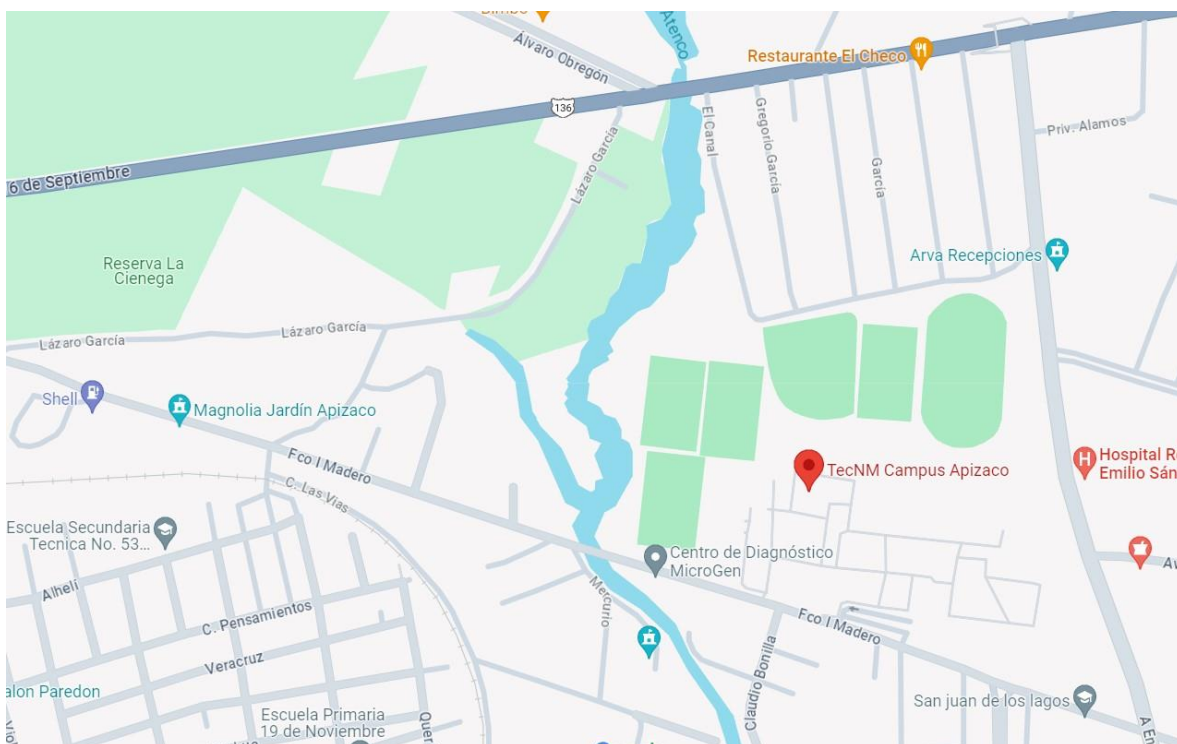
Código Postal: 90300

Teléfono: 241 417 2010

Correo: direccion@apizaco.tecnm.mx

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

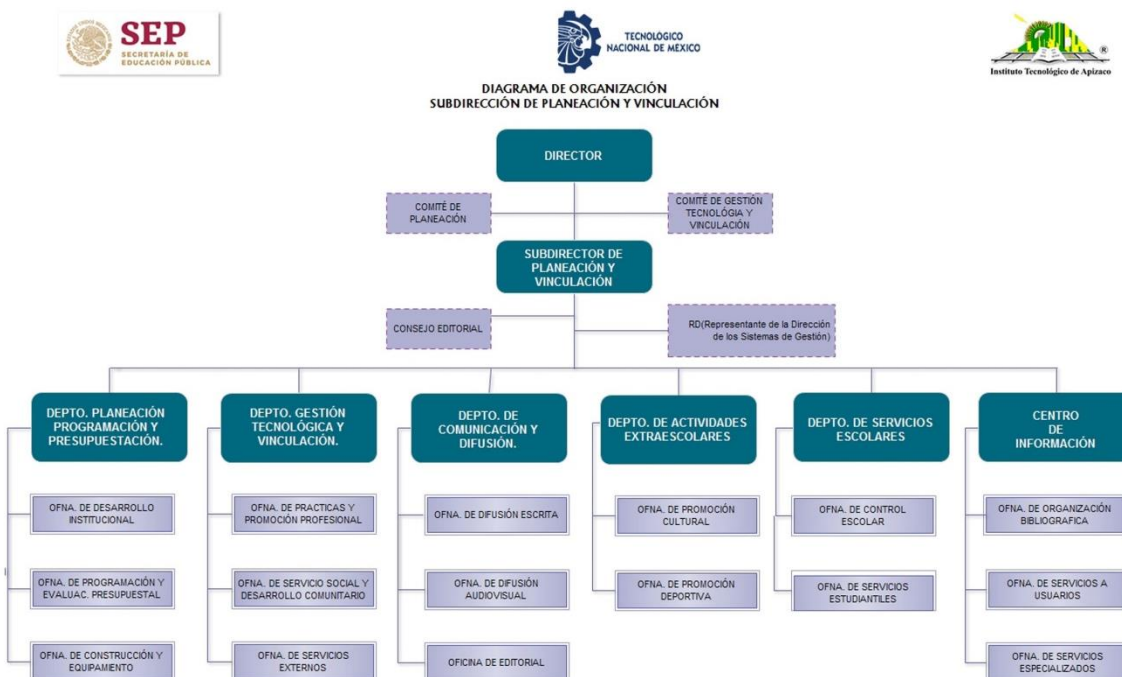
4.1. Mapa de la empresa



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.

Existen varios desafíos y problemas que un proyecto de este tipo podría abordar, uno de estos es la variabilidad de las huellas dactilares: Las huellas dactilares pueden variar significativamente en términos de calidad, tamaño, orientación y nivel de detalle. Esto puede dificultar la extracción y selección de características consistentes y confiables para su identificación, lo cual está relacionado con el ruido en las imágenes. Las imágenes de huellas dactilares pueden estar sujetas a diversos tipos de ruido, como manchas, distorsiones o artefactos de captura. Estos elementos pueden afectar la precisión del proceso de extracción de características y, por lo tanto, la precisión del reconocimiento.

Otro problema importante a tener en cuenta es la eficiencia computacional. El procesamiento de grandes volúmenes de imágenes de huellas dactilares puede requerir una cantidad significativa de recursos computacionales. Optimizar los algoritmos de extracción y selección de características para mejorar la eficiencia sin comprometer la precisión es un desafío importante.

La robustez contra ataques es otro problema importante a tener en cuenta ya que los sistemas de reconocimiento de huellas dactilares pueden ser vulnerables a ataques, como la falsificación de huellas dactilares o la presentación de imágenes manipuladas. Es crucial desarrollar técnicas robustas que puedan detectar y mitigar estos tipos de amenazas.

La interoperabilidad entre diferentes sistemas de identificación biométrica y la compatibilidad con estándares internacionales son aspectos importantes a considerar para garantizar la integración efectiva del sistema en diversos entornos y aplicaciones, sin olvidarnos de la privacidad y protección de datos. La seguridad y la privacidad de los datos biométricos, especialmente de las huellas dactilares, son preocupaciones importantes. Es esencial implementar medidas adecuadas de protección de datos y garantizar el cumplimiento de regulaciones y normativas de privacidad.

Abordar estos problemas requiere un enfoque multidisciplinario que combine conocimientos en procesamiento de imágenes, aprendizaje automático, seguridad de la información y ética en la tecnología. Un proyecto de análisis y experimentación para extracción y selección de características para huellas dactilares puede contribuir significativamente a superar estos desafíos y mejorar la eficacia de los sistemas de identificación biométrica basados en huellas dactilares.

6. Objetivos

6.1. Objetivo General

Desarrollar un sistema robusto y eficiente que permita la extracción y selección óptima de características en huellas dactilares, mediante el análisis exhaustivo de algoritmos y técnicas de procesamiento de imágenes. Este sistema busca mejorar la precisión y confiabilidad de los sistemas de identificación biométrica basados en huellas dactilares, superando desafíos como la variabilidad de las huellas, el ruido en las imágenes y la eficiencia computacional, con el fin de contribuir al avance en la tecnología de reconocimiento biométrico y su aplicación en diversos campos.

6.2. Objetivos Específicos:

- ✚ Investigar y revisar exhaustivamente el estado del arte en técnicas de extracción y selección de características para huellas dactilares, identificando los enfoques más prometedores y las áreas de mejora.
- ✚ Desarrollar algoritmos y metodologías para la extracción de características de huellas dactilares, considerando la variabilidad inherente en las imágenes y la robustez contra posibles ataques y ruido.
- ✚ Evaluar y comparar diferentes métodos de selección de características, analizando su impacto en la precisión y eficiencia del sistema de identificación biométrica.
- ✚ Diseñar un entorno experimental robusto para la evaluación de los algoritmos propuestos, incluyendo conjuntos de datos representativos y métricas de evaluación adecuadas.
- ✚ Implementar un sistema de extracción y selección de características en un entorno de desarrollo adecuado, utilizando herramientas como Python y Visual Basic, y aprovechando bibliotecas y frameworks relevantes.
- ✚ Realizar experimentos controlados para evaluar el rendimiento del sistema en términos de precisión, velocidad y robustez contra diferentes tipos de variabilidad y ataques.
- ✚ Optimizar los algoritmos y parámetros del sistema en función de los resultados experimentales, buscando mejorar continuamente su desempeño y eficiencia.
- ✚ Validar la efectividad y aplicabilidad del sistema mediante pruebas en escenarios reales o simulados, asegurando su viabilidad para su integración en sistemas de identificación biométrica existentes.

7. Justificación

La detección y reconocimiento de rostros y huellas dactilares es una tecnología con múltiples aplicaciones en diversos campos, desde la seguridad y la justicia hasta la gestión de identidades y la privacidad. El reconocimiento facial y de huellas dactilares se utiliza ampliamente en la seguridad pública para la identificación y captura de criminales. Un proyecto en este ámbito contribuiría a mejorar la seguridad ciudadana y a la prevención del delito. La tecnología de reconocimiento facial y de huellas dactilares puede ser invaluable en la identificación de personas desaparecidas, lo que ayudaría a reunir a familias separadas y resolver casos de larga data. Además, En entornos corporativos, gubernamentales y de atención médica, el reconocimiento facial y de huellas dactilares facilita la gestión de identidades y el control de accesos, mejorando la seguridad y la eficiencia operativa. También la tecnología de reconocimiento facial se utiliza para crear experiencias de usuario más convenientes, como el desbloqueo de dispositivos móviles y la autenticación en aplicaciones, lo que mejora la usabilidad y la seguridad.

8. Organigrama

Cronograma de Actividades

ACTIVIDAD		SEPTIEMBRE			OCTUBRE			NOVIEMBRE			DICIEMBRE			ENERO		
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Inicio	P															
	R															
Introducción	P															
	R															
Asignación del Proyecto	P															
	R															
Definición de requisitos y alcance	P															
	R															
Planificación del Proyecto	P															
	R															
Diseño del Sistema	P															
	R															
Desarrollo del software	P															
	R															
Entrenamiento del modelo de reconocimiento huellas	P															
	R															

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, dermatoglifo o dactilograma (del griego δάκτυλος "dedo" y γράμμα "dibujo") (en algunos lugares, también huella digital) es la impresión visible o moldeada que produce el contacto de las crestas papilares de un dedo de la mano (generalmente se usan el dedo pulgar o el dedo índice) sobre una superficie. La primera técnica de identificación de personas mediante estas fue inventada por el francés Alphonse Bertillon, y luego mejorada y puesta por primera vez en práctica por el nacionalizado argentino Juan Vucetich.¹



Es una característica individual que se utiliza como medio de identificación de las personas (ver biometría). Se clasifican por sus características en:

Visibles o positivas: Son las que dejan los dedos al estar impregnados de algún colorante, este material puede ser sangre, tinta, polvo o cualquier otra sustancia con la que puedan quedar marcadas las crestas papilares y puedan ser observadas a simple vista.

Moldeadas: Son las que aparecen impresas en forma de molde, estas se marcan en materia plástica, como la grasa, jabón, plastilina, etcétera.

Naturales: Aparecen de forma natural en los pulpejos de ambas manos, desde los seis meses de vida intrauterina hasta la muerte e incluso en el proceso de putrefacción.

Artificiales: Son aquellas que se encuentran plasmadas en forma intencional con alguna sustancia, esencialmente con tinta para su estudio.

La disciplina científica que estudia las huellas dactilares se llama dactiloscopía, y dentro de ella existen dos grandes ramas con su propia clasificación de huellas.

9.2. Los fundamentos de la identificación de huellas dactilares

Terminología de identificación de huellas dactilares

Las huellas dactilares son extremadamente complejas. Se utilizan características definitorias, muchas de las cuales han sido establecidas por organismos encargados de hacer cumplir la ley, para “leer” y clasificar las huellas dactilares. Aunque las empresas de biometría como DigitalPersona no guardan imágenes de huellas dactilares y no utilizan el mismo proceso manual para analizarlas, muchas de las mismas metodologías establecidas a lo largo de los años en las fuerzas del orden se utilizan para nuestros algoritmos digitales. Los sistemas biométricos autentican a los usuarios comparando las crestas y patrones del dedo. Para desglosarlo aún más, el software busca distinciones dentro de estas áreas:

Crestas

La piel de las superficies internas de las manos, dedos de las manos, de los pies y de los pies tiene “surcos” o está cubierta con patrones elevados concéntricos. Estas crestas se llaman crestas de fricción y proporcionan fricción, lo que nos facilita agarrar y sujetar objetos y superficies sin resbalar. Son las muchas diferencias en la forma en que las crestas de fricción están modeladas, rotas y bifurcadas las que hacen que las áreas de la piel con crestas, incluidas las huellas dactilares, sean distintivas.

Características globales versus locales

Se utilizan dos tipos de características dactilares en la identificación de individuos: características globales y características locales. Las características globales son aquellas características que se pueden ver a simple vista e incluyen:

-  Área de patrón
-  Área central
-  Tipo de líneas
-  Delta
-  Conteo de crestas básico
-  Patrones de cresta

Las características locales se conocen como puntos minuciosos. Son las pequeñas características de las crestas de las huellas dactilares. Su disposición bidimensional es distintiva y se utiliza para el reconocimiento. Es posible que dos o más individuos tengan características globales similares, pero aun así tengan huellas dactilares diferentes y distintivas porque las características locales, es decir, la disposición bidimensional de los puntos minuciosos, es diferente.

9.3. Funciones globales

Área de patrón: el área de patrón es la parte de la huella digital que contiene las características globales. Las huellas dactilares se leen y clasifican según la información del área de patrón. Ciertos puntos minuciosos que se utilizan para el reconocimiento final pueden estar fuera del área del patrón.



Punto central: el punto central, ubicado aproximadamente en el centro de la impresión del dedo, se utiliza como punto de referencia inicial para leer y clasificar la impresión.

Líneas de tipo: las líneas de tipo son las dos crestas más internas que comienzan paralelas, divergen y rodean o tienden a rodear el área del patrón. Cuando hay una ruptura definitiva en una línea tipo, la cresta inmediatamente fuera de esa línea se considera su continuación.

Delta: el delta es el punto en la primera bifurcación (donde la cresta se bifurca en dos direcciones diferentes), cresta final abrupta, encuentro de dos crestas, punto, cresta fragmentaria o cualquier punto sobre una cresta en o más cerca del centro de divergencia de dos líneas tipográficas. El delta está situado directamente delante del punto de divergencia de la línea. Es un punto fijo definido que se utiliza para facilitar el recuento y rastreo de crestas.



Recuento de crestas: el recuento de crestas suele ser el número de crestas entre el delta y el núcleo. Para establecer el recuento de crestas, se traza una línea imaginaria desde el delta hasta el núcleo; Se cuenta cada cresta que toca esta línea.

9.4. Patrones básicos de crestas

Para facilitar la búsqueda de huellas dactilares en grandes bases de datos de huellas dactilares, los expertos las clasifican en grupos según los patrones en las crestas. Estas agrupaciones o patrones básicos de crestas no son suficientes por sí solos para la identificación, pero ayudan a acotar la búsqueda y acelerar el tiempo de procesamiento. Una vez que se identifica una huella digital como un grupo particular, como un verticilo, la búsqueda solo continúa comparando la huella con todos los demás tipos de verticilos en la base de datos e ignora las otras agrupaciones.

Se han definido una serie de agrupaciones básicas de patrones de crestas. Tres de los más comunes son bucle, arco y verticilo.

1. BUCLE

El bucle es el tipo más común de patrón de huellas dactilares y representa aproximadamente el 65% de todas las huellas dactilares.



2. ARCO

El patrón del arco es una curva más abierta que el bucle. Hay dos tipos de patrones de arco: el arco simple y el arco en forma de carpa.



3. VERTICULO

Los patrones de verticilos ocurren en aproximadamente el 30% de todas las huellas dactilares y están definidos por al menos una cresta que forma un círculo completo.



Ciertos productos biométricos basan la identificación en la correlación de patrones de crestas globales, o en la comparación de una imagen de patrón de huellas dactilares con otra. DigitalPersona cree que el reconocimiento de huellas dactilares de alta calidad Los algoritmos deben ir un paso más allá y hacer que el algoritmo se base en puntos minuciosos además de características globales.

9.5. Puntos minuciosos

Las crestas de las huellas dactilares no son crestas rectas y continuas. En cambio, se rompen, bifurcan, interrumpen o cambian de dirección. Los puntos en los que las crestas terminan, se bifurcan y cambian se denominan puntos minuciosos que proporcionan información distintiva e identificativa.

Hay cinco características de los puntos minuciosos en huellas dactilares:

1. Tipo



Hay varios tipos de puntos minuciosos. Los más comunes son las terminaciones de crestas y las bifurcaciones de crestas.

Fin de cresta: ocurre cuando una cresta termina abruptamente.

Bifurcación de cresta: el punto en el que una cresta se divide en ramas.

2. Orientación

El punto de la cresta en el que reside una minucia se llama orientación del punto de minucia.

3. Frecuencia espacial

La frecuencia espacial se refiere a qué tan separadas están las crestas en relación con el punto minucioso.

4. Curvatura

La curvatura se refiere a la tasa de cambio de la orientación de las crestas.

5. Posición

La posición del punto minucioso se refiere a su ubicación, ya sea en un sentido absoluto o relativo a puntos fijos como el delta y los puntos centrales.

9.6. ¿Son las huellas dactilares realmente distintivas y persistentes?

Detrás de todos los métodos de reconocimiento de huellas dactilares está la afirmación de que no hay dos huellas dactilares iguales. La creencia de DigitalPersona en esta afirmación se basa en dos principios fundamentales:

Teórico: Sabemos cuánta información se incluye en una huella digital y podemos crear modelos estadísticos en torno a esto. Hay hasta 70 puntos minuciosos en cada impresión, y cada uno de estos puntos tiene varias características como se describe anteriormente.

La posibilidad de encontrar conjuntos de puntos minuciosos que sean similares con respecto a estas características es tan pequeña que resulta insignificante.



Empírico: FBI Sólo los archivos contienen más de 200 millones de huellas dactilares. En todos los datos que se han recopilado durante los últimos cien años, utilizando las metodologías de clasificación descritas anteriormente, nunca ha habido Se encontraron dos huellas dactilares que eran idénticas.

El carácter distintivo de las huellas dactilares está claramente establecido. La persistencia biológica subyacente de las huellas dactilares también es un hecho bien establecido, informado en varios estudios de huellas dactilares realizados en diferentes campos científicos durante el siglo pasado. Sin embargo, existen algunos factores que afectan a la

persistencia de las huellas dactilares con respecto a los sistemas de reconocimiento automático de huellas dactilares:

Los motores de reconocimiento automático de huellas dactilares funcionan con imágenes de huellas dactilares. La imagen se adquiere mediante la interacción de un dedo con un lector de huellas dactilares. La calidad de la imagen de la huella dactilar se ve directamente afectada por el mecanismo que se utiliza para recoger la huella. Incluso se crean huellas dactilares con tinta sobre papel; El grado de presión, la cantidad de tinta y otros factores que no tienen nada que ver con la huella dactilar en sí pueden afectar la información recopilada. Lo mismo se aplica a los mecanismos electrónicos de recaudación. Por ejemplo, un lector de huellas dactilares puede estar sucio y, en consecuencia, la imagen de la huella dactilar adquirida puede tener artefactos que contribuyan a reducir la persistencia.

Las características subyacentes de las huellas dactilares biológicas no cambian a lo largo de la vida, salvo que queden cicatrices de una lesión importante en el dedo. Sin embargo, las condiciones de la piel pueden cambiar con el tiempo según el clima, la ocupación, el estilo de vida, el género, la raza y las actividades del individuo.

Aun así, incluso con estas advertencias, las huellas dactilares son datos biométricos extremadamente confiables. Si bien la coincidencia de huellas dactilares no es perfecta, el motor de reconocimiento de huellas dactilares de DigitalPersona es el mejor de su clase. Motor de coincidencia de huellas dactilares altamente preciso, rápido y confiable.

10. Herramientas Tecnológicas Usadas

10.1. Lector biométrico U.are.U 4500 Reader de DigitalPersona

El lector biométrico U.are.U 4500 Reader de DigitalPersona es un dispositivo diseñado para la captura precisa y rápida de huellas dactilares con fines de identificación y autenticación biométrica. DigitalPersona, 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. VisualBasic

Utilizamos VisualBasic 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 se destaca por su enfoque en la programación orientada a eventos y su facilidad de aprendizaje. Está diseñado para permitir a los desarrolladores crear aplicaciones de manera rápida y eficiente mediante la utilización de una interfaz gráfica de usuario (GUI) intuitiva y una sintaxis simple y legible.

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.
- Posibilita añadir soporte para ejecución de scripts, VBScript o JScript, en las aplicaciones mediante Microsoft Script Control.⁷
- Tiene acceso a la API multimedia de DirectX (versiones 7 y 8). También está disponible, de forma no oficial, un componente para trabajar con OpenGL 1.1.8
- Existe una versión, VBA, integrada en las aplicaciones de Microsoft Office, tanto Windows como Mac, que permite programar macros para extender y automatizar funcionalidades en documentos, hojas de cálculo y bases de datos (Access).
- Si bien permite desarrollar grandes y complejas aplicaciones, también provee un entorno adecuado para realizar pequeños prototipos rápidos.

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

10.3. Bibliotecas DLL

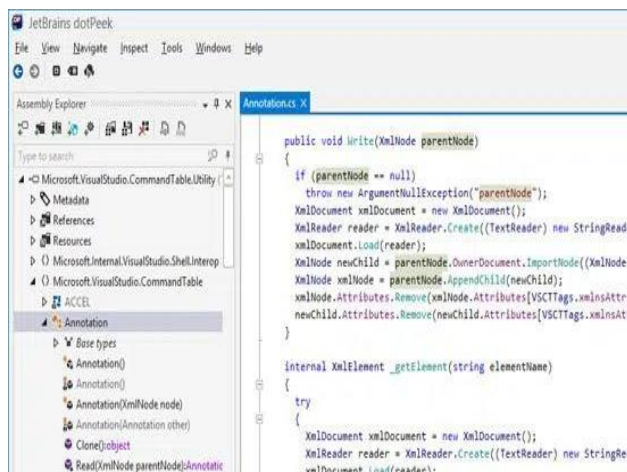
Utilizamos las bibliotecas DLL como un método de comunicación entre VisualBasic y Python, exportamos nuestro proyecto de VisualBasic a una biblioteca DLL la cual podremos utilizar todos sus métodos en Python mediante Python.NET.

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.

10.4. 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.

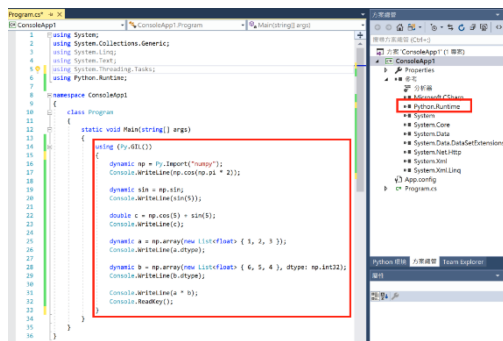
Python usa tipado dinámico y conteo de referencias para la gestión de memoria.

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.



10.5. 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).

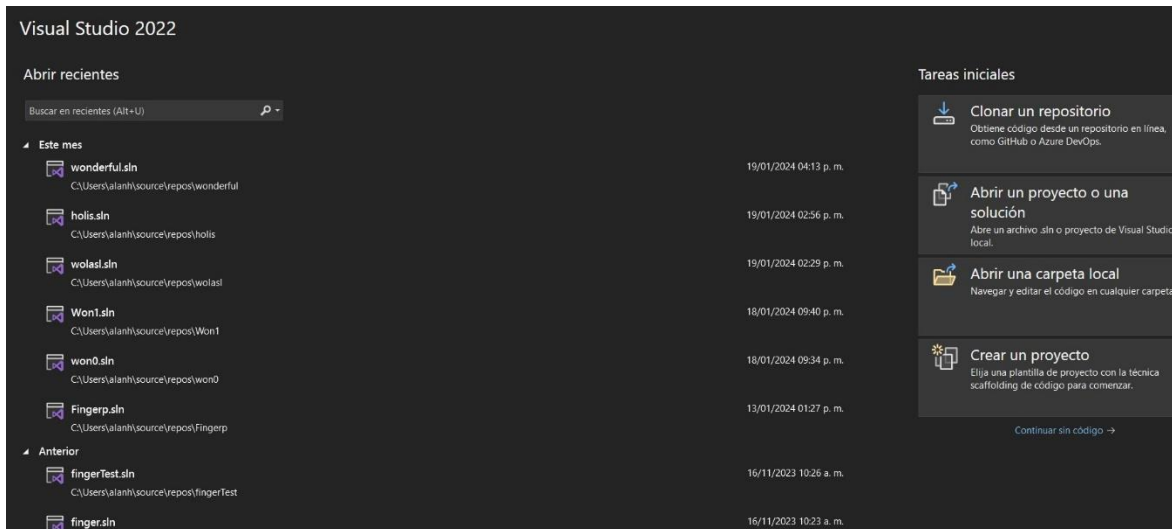
Tenga en cuenta que este paquete no implementa Python como lenguaje CLR de primera clase; no produce código administrado (IL) a partir del código Python. Más bien, es una integración del motor CPython con el tiempo de ejecución .NET o Mono. Este enfoque le permite utilizar servicios CLR y continuar usando 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 .

10.6. VisualStudio 2022

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:

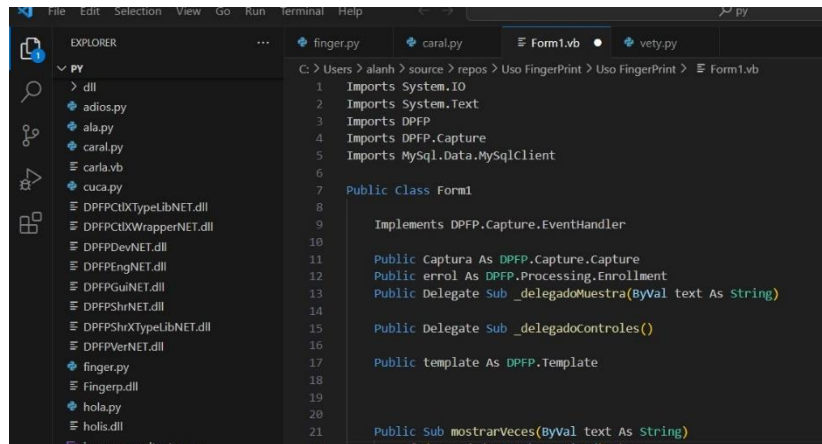
- **Soporte multiplataforma:** Visual Studio admite el desarrollo de aplicaciones para una variedad de plataformas, incluyendo Windows, iOS, Android y Linux. Esto permite a los desarrolladores crear aplicaciones para una amplia gama de dispositivos y sistemas operativos.
- **Amplia gama de lenguajes 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.

10.7. VisualStudio 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.

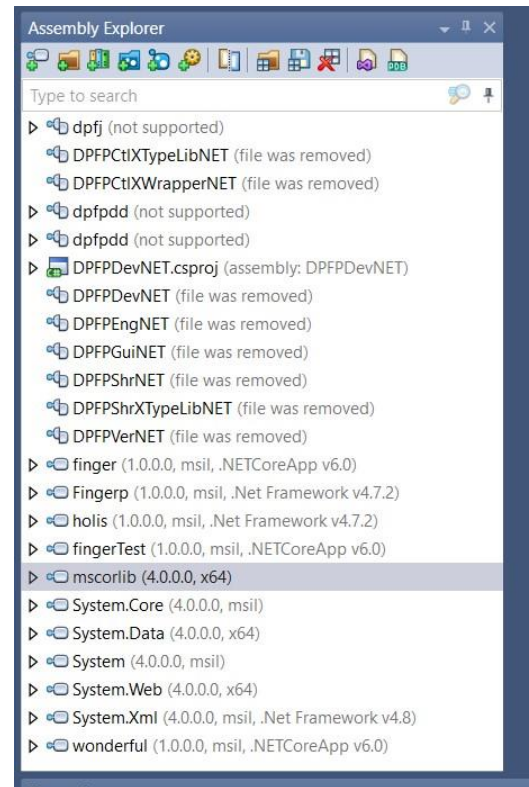
En el rol de editor de código fuente, Visual Studio Code permite cambiar la página de códigos en la que se guarda el documento activo, el carácter que identifica el salto de línea (una opción entre LF y CRLF) y el lenguaje de programación del documento activo.

10.8. 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.

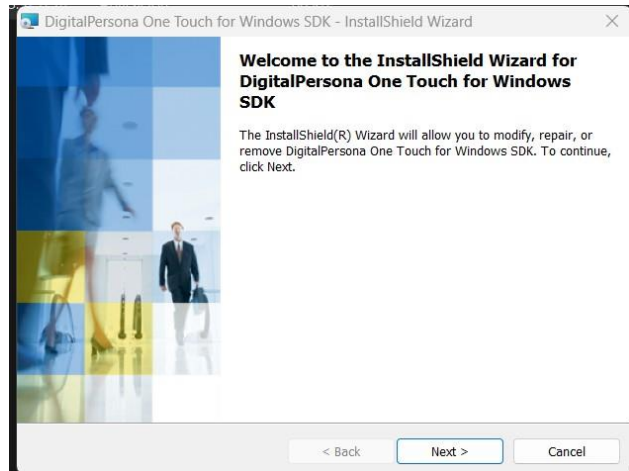
Funcionalidad de búsqueda: dotPeek proporciona una función de búsqueda que permite a los desarrolladores buscar palabras clave específicas dentro del código descompilado. Esto facilita la ubicación de secciones relevantes de código y ayuda a los desarrolladores a encontrar rápidamente lo que están buscando.

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.

```
<?xml version="1.0" encoding="utf-8"?>
<Project DefaultTargets="Build" xmlns="http://schemas.microsoft.com/developer/msbuild/2003" Tool:
<!--Project was exported from assembly: C:\Users\alanh\OneDrive\Documentos\digitalpersonal\DPFI
<PropertyGroup>
  <Configuration Condition="'$(Configuration)' == ''">Debug</Configuration>
  <Platform Condition="'$(Platform)' == ''">AnyCPU</Platform>
  <ProjectGuid>{1BA5F731-E336-47E0-AAFC-3F81D70A890A}</ProjectGuid>
  <OutputType>Library</OutputType>
  <AssemblyName>DPFPDevNET</AssemblyName>
  <ApplicationVersion>1.6.1.0</ApplicationVersion>
  <RootNamespace>DPFP</RootNamespace>
  <FileUpgradeFlags>
  </FileUpgradeFlags>
  <UpgradeBackupLocation>
  </UpgradeBackupLocation>
  <OldToolsVersion>2.0</OldToolsVersion>
</PropertyGroup>
<PropertyGroup Condition="'$(Configuration)|$(Platform)' == 'Debug|AnyCPU' ">
  <PlatformTarget>AnyCPU</PlatformTarget>
  <DebugSymbols>true</DebugSymbols>
  <DebugType>full</DebugType>
  <Optimize>>false</Optimize>
  <OutputPath>bin\Debug\</OutputPath>
  <DefineConstants>DEBUG;TRACE</DefineConstants>
  <ErrorReport>prompt</ErrorReport>
  <WarningLevel>4</WarningLevel>
</PropertyGroup>
<PropertyGroup Condition="'$(Configuration)|$(Platform)' == 'Release|AnyCPU' ">
  <PlatformTarget>AnyCPU</PlatformTarget>
  <DebugType>pdbonly</DebugType>
```

10.9. SDK DigitalPersona

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

Funcionalidades de las librerías del SDK:

Folder	File	Description
<destination folder>\Bin	DPCOper2.dll DPDevice2.dll DPDevTS.dll DpHostW.exe DPmsg.dll DPMux.dll DpSvInfo2.dll DPTSCInt.dll DPCrStor.dll	DLLs and executable file used by the all of the One Touch for Windows APIs
GlobalAssemblyCache	DPFPShrNET.dll DPFPDevNET.dll DPFPEngNET.dll DPFPVerNET.dll DPFPGuiNET.dll DPFPCTIXLib.dll DPFPCTIXTypeLibNET.dll DPFPCTIXWrapperNET.dll DPFPShrXTypeLibNET.dll	DLLs used by the One Touch for Windows: .NET Edition API
<system folder>	DPFPApi.dll DpClback.dll dpHFtrEx.dll dpHMatch.dll DPFpUI.dll	DLLs used by all of the One Touch for Windows APIs

10.10.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.



En junio de 2008 fue anunciado que la recién formada Django Software Foundation se haría cargo de Django en el futuro.

La meta fundamental de Django es facilitar la creación de sitios web complejos. Django pone énfasis en el re-uso, la conectividad y extensibilidad de componentes, el desarrollo rápido y el principio «DRY» (del inglés Don't Repeat Yourself, «No te repitas»). El lenguaje Python es usado en todos los componentes del framework, incluso en configuraciones, archivos, y en sus modelos de datos.

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.

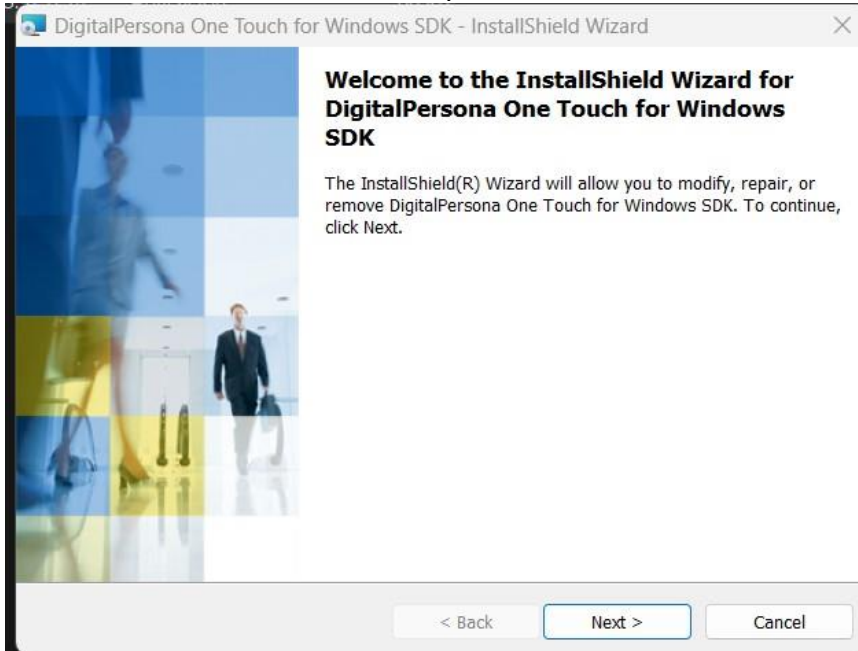
Características

-  **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.

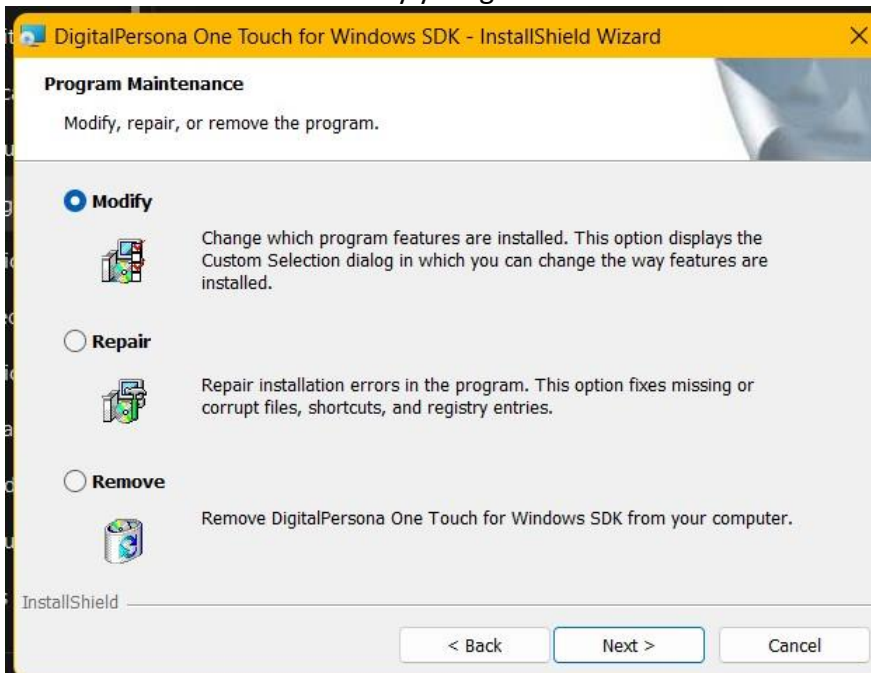
11. Desarrollo del Proyecto

11.1. Instalación del SDK de DigitalPersona para sensor biométrico

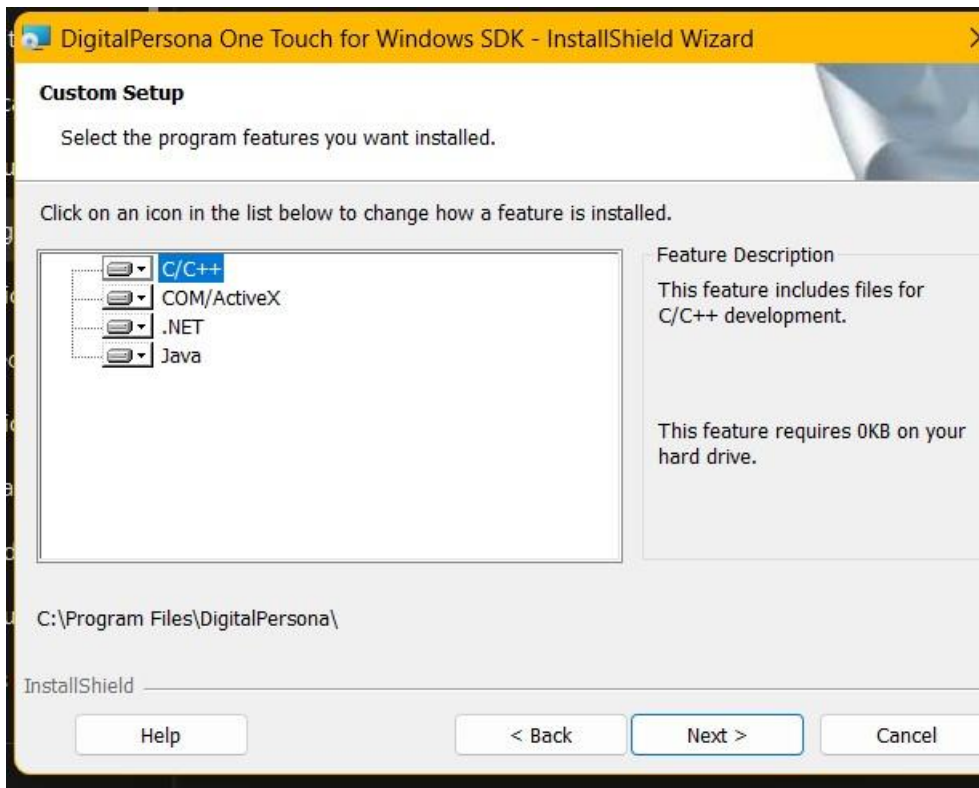
1. Abrimos el instalador SDK y hacemos clic en Next



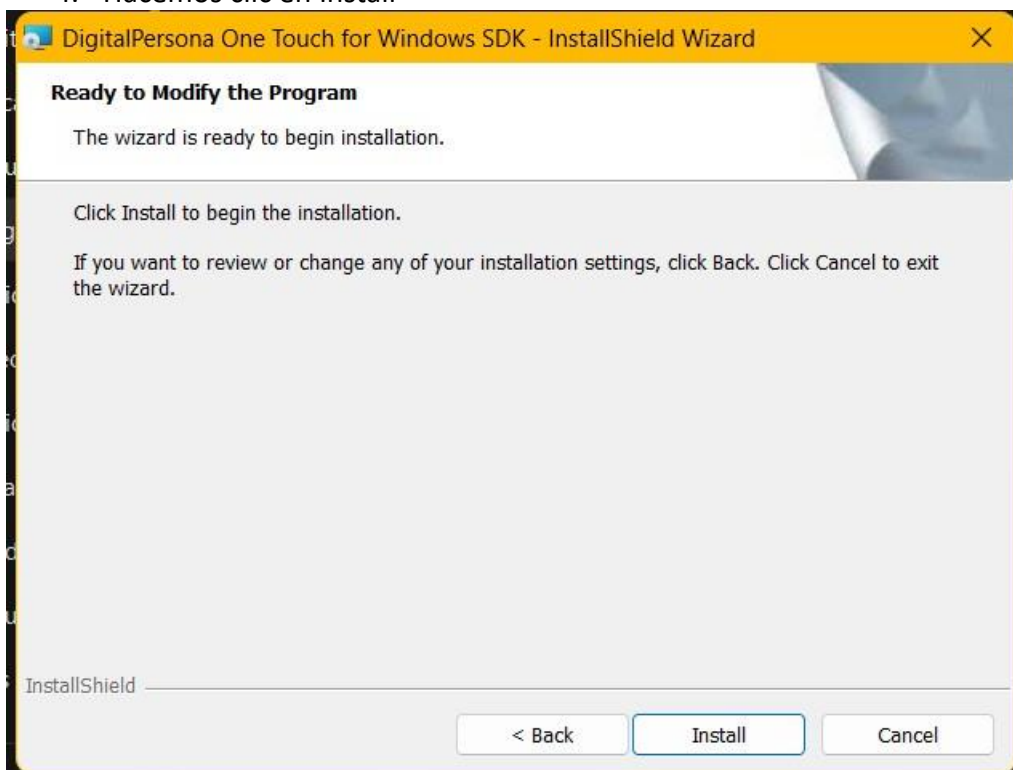
2. Hacemos clic en Modify y luego en Next



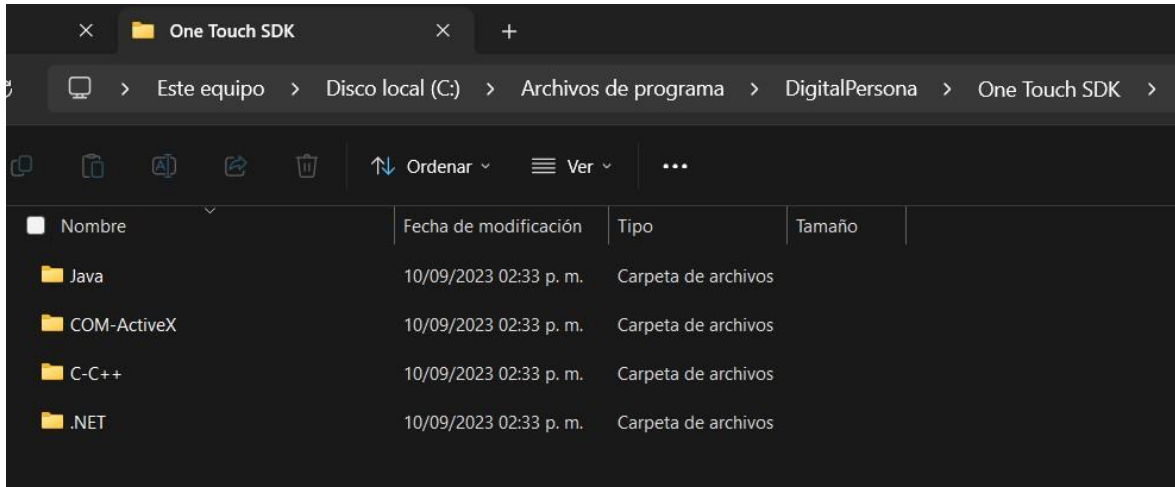
3. Hacemos clic en Next



4. Hacemos clic en Install

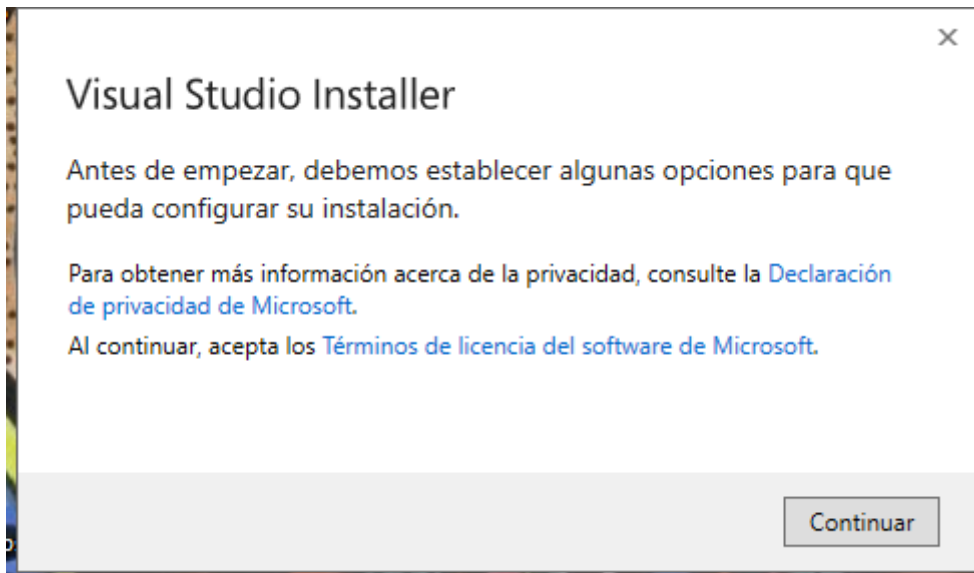


5. Si nos dirigimos a la carpeta Archivos de Programa/DigitalPerona podremos ver los archivos del SDK

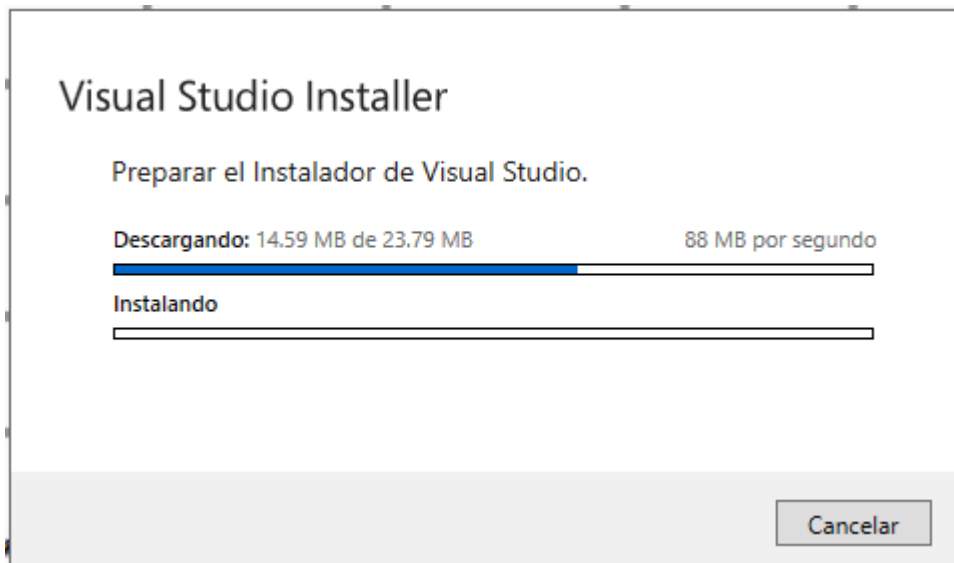


11.2. Instalación de VisualStudio

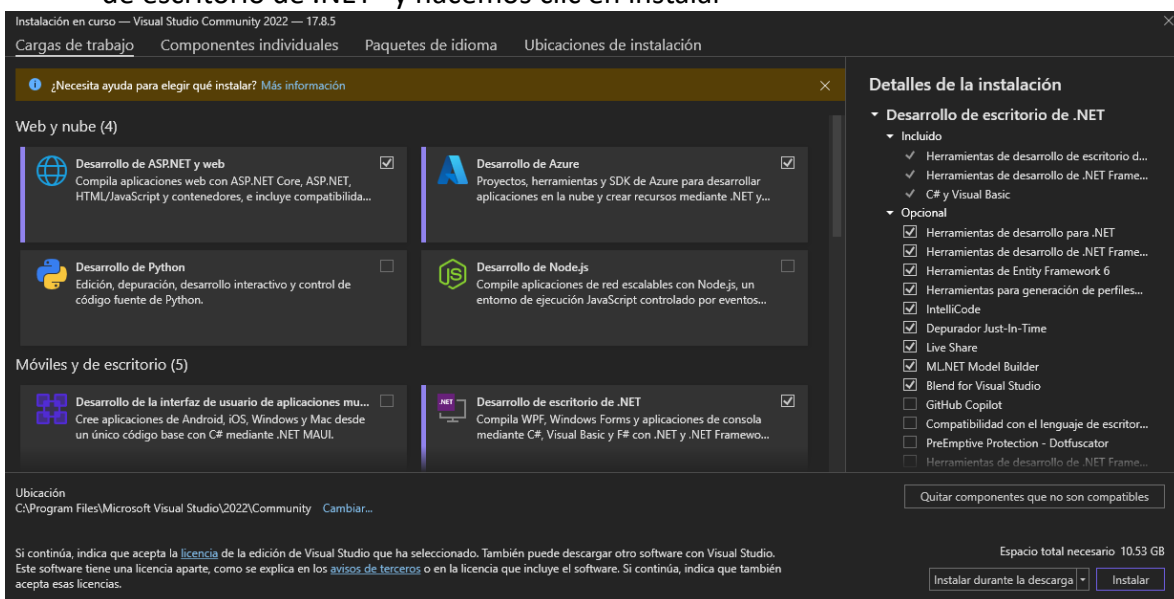
1. Hacemos clic en continuar



2. Esperamos a que termine la descarga

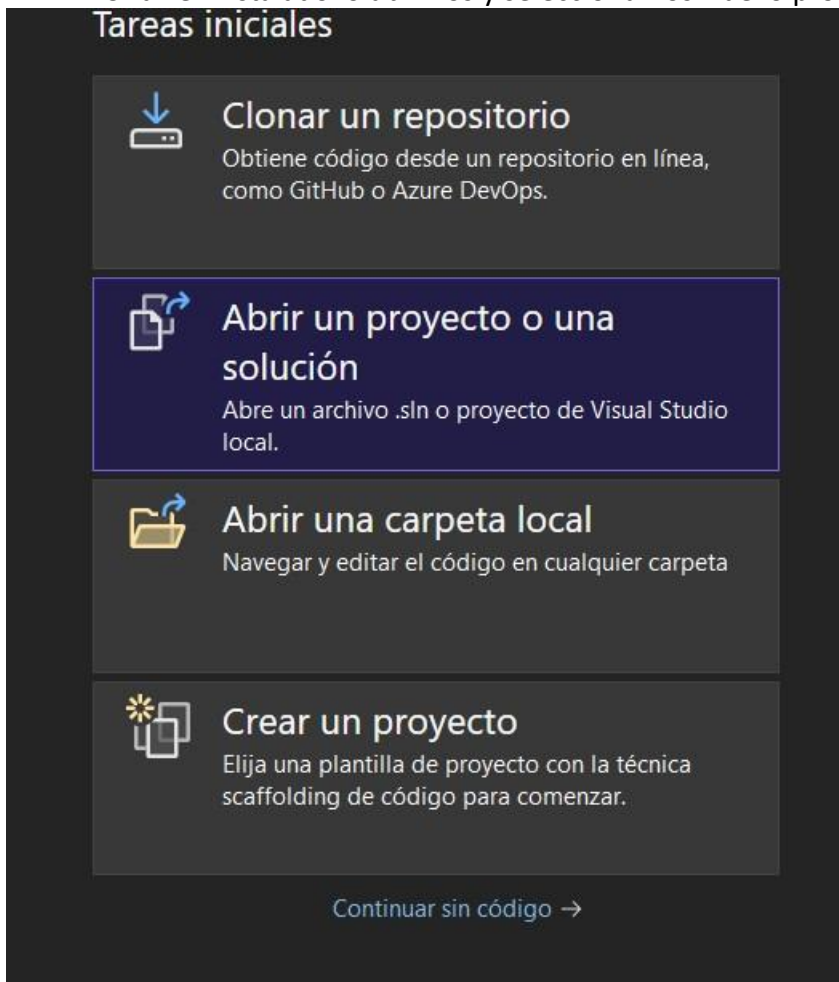


3. Seleccionamos “Desarrollo de ASP.NET y web”, “Desarrollo de Azure” y “Desarrollo de escritorio de .NET” y hacemos clic en instalar

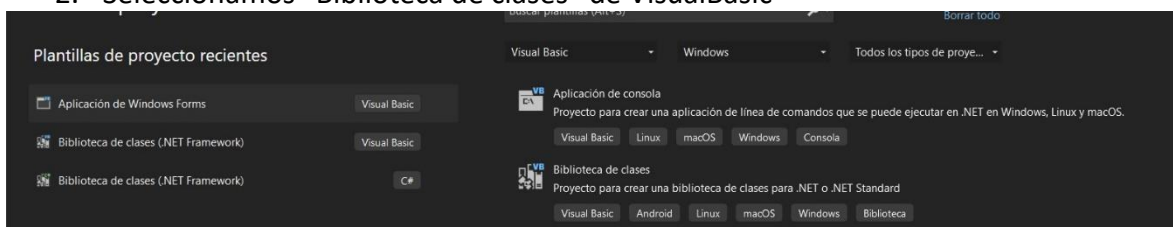


11.3. Creación de la Biblioteca DLL

1. Una vez instalado lo abrimos y seleccionamos Nuevo proyecto



2. Seleccionamos “Biblioteca de clases” de VisualBasic



3. Elegimos un nombre de proyecto y hacemos clic en Siguiente

Configure su nuevo proyecto

Biblioteca de clases (.NET Framework) Visual Basic Windows Biblioteca

Nombre del proyecto

ClassLibrary1

Ubicación

C:\Users\alanh\source\repos

Nombre de la solución ⓘ

ClassLibrary1

☐ Colocar la solución y el proyecto en el mismo directorio

Marco

.NET Framework 4.7.2

Proyecto se creará en "C:\Users\alanh\source\repos\ClassLibrary1\ClassLibrary1\"

4. Hacemos clic en crear

Información adicional

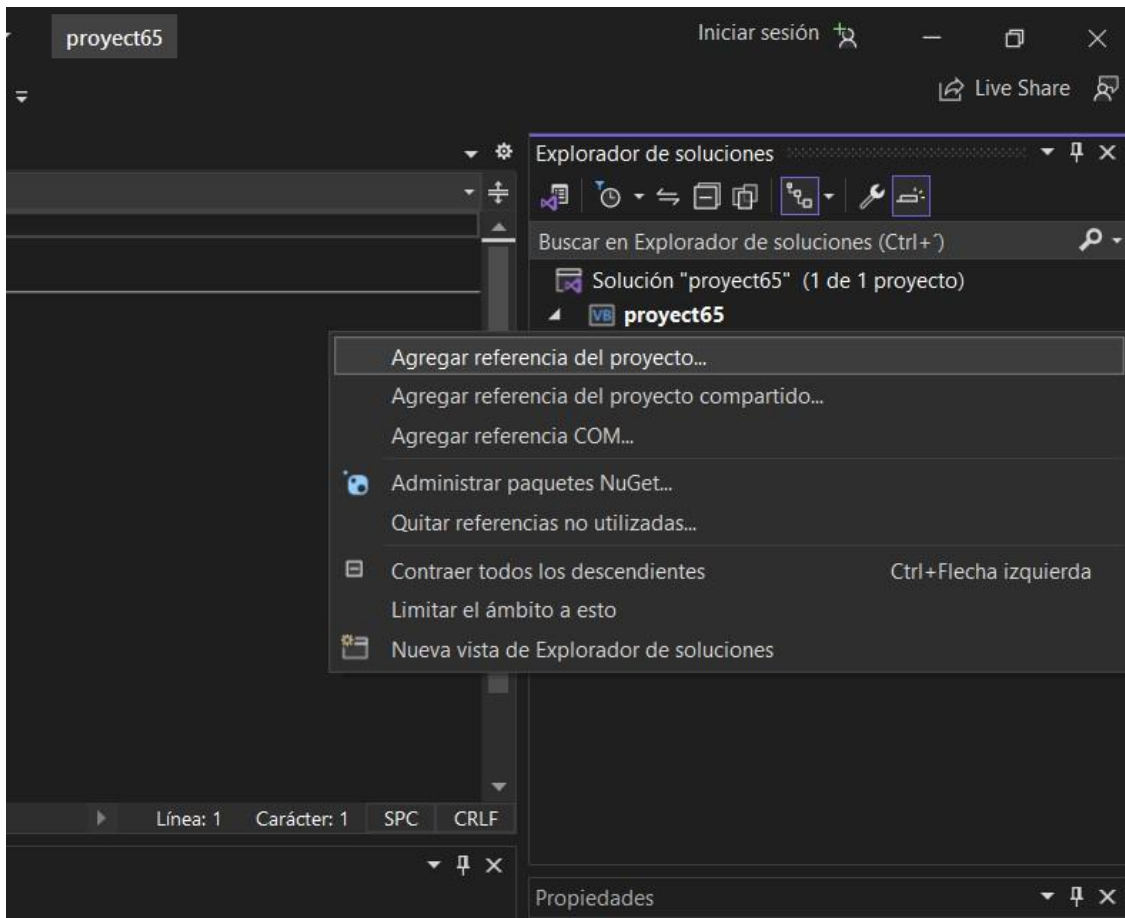
Biblioteca de clases Visual Basic Android Linux macOS Windows Biblioteca

Framework ⓘ

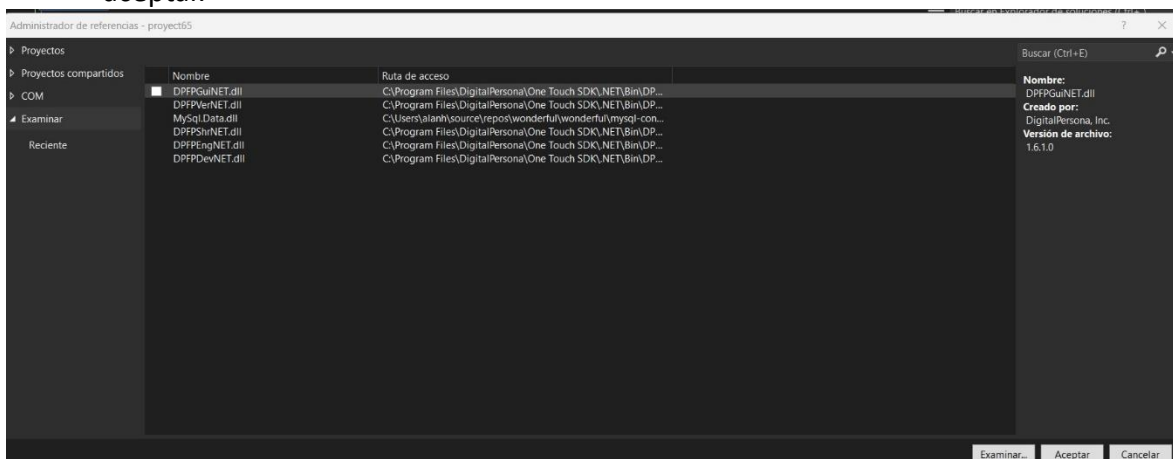
.NET 6.0 (Compatibilidad a largo plazo)

Atrás Crear

- Una vez creado el proyecto, en el menú de explorador de soluciones, hacemos clic derecho en el nombre de nuestro proyecto y luego agregar referencia del proyecto.



6. Buscamos las librerías DLL necesarias en la carpeta .NET del SDK (C:\Program Files\DigitalPersona\One Touch SDK\.NET\Bin), las seleccionamos y hacemos clic en aceptar.



7. Código de VisualBasic
 - a. Primero se importan las referencias necesarias
 - b. Se declaran las variables de la captura
 - c. Se crea el constructor de la biblioteca
 - d. Se crean los métodos de iniciar captura y detener captura

```

1 Imports System.Data.SqlClient
2 Imports System.IO
3 Imports System.Linq.Expressions
4 Imports System.Text
5 Imports System.Text.RegularExpressions
6 Imports System.Windows.Forms
7 Imports DPFP
8 Imports DPFP.Capture
9 Imports Microsoft.SqlServer.Server
10 Imports MySql.Data.MySqlClient
11
12 Public Class Class1
13     Implements DPFP.Capture.EventHandler
14
15     Private Captura As DPFP.Capture.Capture
16     Private Enroler As DPFP.Processing.Enrollment
17     Private Delegate Sub _delegadoMuestra(ByVal text As String)
18     Private Delegate Sub _delegadoControles()
19     Private template As DPFP.Template
20
21
22
23
24     Protected Overridable Sub Init()
25         Try
26             Captura = New DPFP.Capture.Capture()
27
28             If Not Captura Is Nothing Then
29                 Captura.EventHandler = Me
30                 Enroler = New DPFP.Processing.Enrollment()
31                 Dim tex As New StringBuilder()
32                 tex.AppendFormat("Necesitas pasar el dedo {0} veces", Enroler.FeaturesNeeded)
33
34             Else
35                 MessageBox.Show("Error fatalisimo")
36             End If
37
38             Catch ex As Exception
39                 MessageBox.Show("Error fatalete")
40             End Try
41         End Sub
42
43     Protected Sub iniciarCaptura()
44         If Not Captura Is Nothing Then
45             Try
46                 Captura.StartCapture()
47             Catch ex As Exception
48                 MessageBox.Show(ex.ToString)
49             End Try
50         End If
51     End Sub
52
53     Protected Sub pararCaptura()
54         If Not Captura Is Nothing Then
55             Try
56                 Captura.StopCapture()
57             Catch ex As Exception
58                 MessageBox.Show("Error fatal al stop captura")
59             End Try
60         End If
61     End Sub

```

8. Métodos de eventos del detector

```

0 referencias
Public Sub OnComplete(Capture As Object, ReaderSerialNumber As String, Sample As Sample) Implements EventHandler.OnComplete
    : MessageBox.Show("EL LECTOR SE HA COMPLETADO")
End Sub

0 referencias
Public Sub OnFingerGone(Capture As Object, ReaderSerialNumber As String) Implements EventHandler.OnFingerGone
    : MessageBox.Show("EL LECTOR SE HA IDO")
End Sub

0 referencias
Public Sub OnFingerTouch(Capture As Object, ReaderSerialNumber As String) Implements EventHandler.OnFingerTouch
    : MessageBox.Show("SJAKSJAKJ")
End Sub

0 referencias
Public Sub OnReaderConnect(Capture As Object, ReaderSerialNumber As String) Implements EventHandler.OnReaderConnect
    : MessageBox.Show("EL LECTOR SE HA CONECTADO")
End Sub

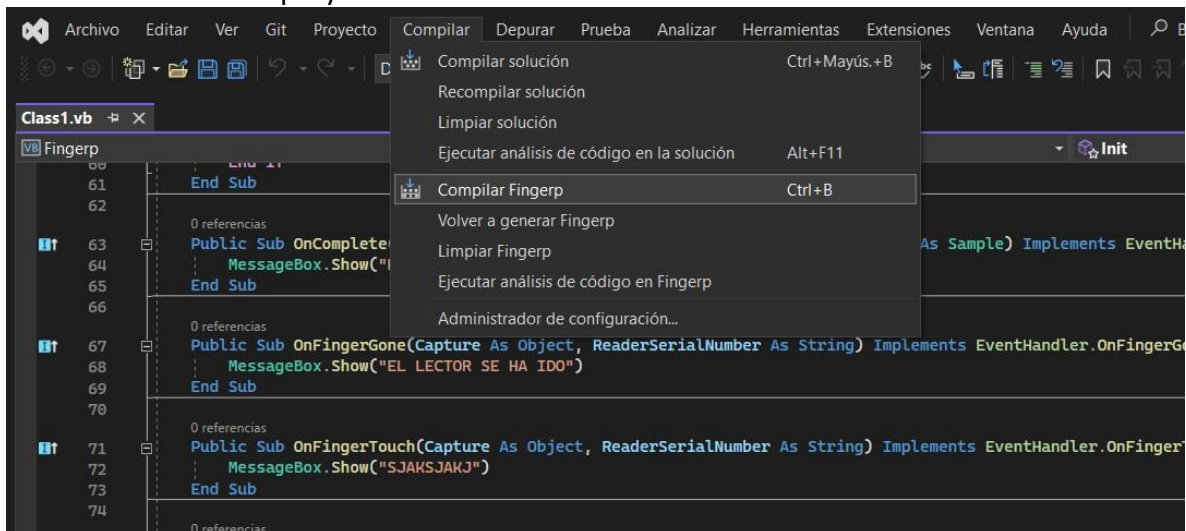
0 referencias
Public Sub OnReaderDisconnect(Capture As Object, ReaderSerialNumber As String) Implements EventHandler.OnReaderDisconnect
    : MessageBox.Show("EL LECTOR SE HA DESCONECTADO")
End Sub

0 referencias
Public Sub OnSampleQuality(Capture As Object, ReaderSerialNumber As String, CaptureFeedback As CaptureFeedback) Implements EventHandler.OnSampleQuality
End Sub

0 referencias
Public Sub go()
    Init()
    iniciarCaptura()
End Sub

```

9. Una vez que el código esta listo, hacemos clic en compilar y luego en “Compilar <Nombre del proyecto>”.

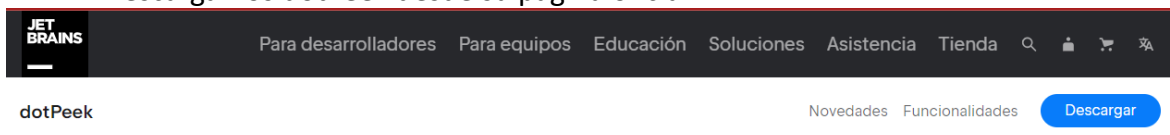


10. Una vez compilado podremos acceder al DLL en la ruta de nuestro proyecto, tendrá el mismo nombre que el nombre del proyecto.

Disco local (C:) > Usuarios > alanh > source > repos > Fingerp > Fingerp > bin > Debug				
Ordenar Ver				
Nombre	Fecha de modificación	Tipo	Tamaño	
DPFPctiXWrapperNET.dll	17/08/2010 05:44 p. m.	Extensión de la ap...	30 KB	
DPFPDevNET.dll	17/08/2010 05:44 p. m.	Extensión de la ap...	42 KB	
DPFPEngNET.dll	17/08/2010 05:44 p. m.	Extensión de la ap...	30 KB	
DPFPGuiNET.dll	17/08/2010 05:44 p. m.	Extensión de la ap...	42 KB	
DPFPShrNET.dll	17/08/2010 05:44 p. m.	Extensión de la ap...	26 KB	
DPFPVerNET.dll	17/08/2010 05:44 p. m.	Extensión de la ap...	26 KB	
Fingerp.dll	16/11/2023 11:18 a. m.	Extensión de la ap...	11 KB	
Fingerp.pdb	16/11/2023 11:18 a. m.	Program Debug D...	42 KB	
Fingerp.xml	16/11/2023 11:18 a. m.	Archivo de origen ...	1 KB	
MySQL.Data.dll	02/10/2023 03:42 p. m.	Extensión de la ap...	1,167 KB	
MySQL.Data.xml	02/10/2023 03:42 p. m.	Archivo de origen ...	976 KB	
System.Configuration.ConfigurationM...	02/10/2023 03:42 p. m.	Extensión de la ap...	372 KB	
System.Runtime.CompilerServices.Uns...	10/09/2023 01:50 p. m.	Extensión de la ap...	18 KB	
System.Security.Cryptography.Protecte...	02/10/2023 03:42 p. m.	Extensión de la ap...	25 KB	
System.Security.Permissions.dll	02/10/2023 03:42 p. m.	Extensión de la ap...	91 KB	
System.Text.Encodings.Web.dll	02/10/2023 03:42 p. m.	Extensión de la ap...	70 KB	
System.Text.Json.dll	02/10/2023 03:42 p. m.	Extensión de la ap...	532 KB	
System.Windows.Extensions.dll	02/10/2023 03:42 p. m.	Extensión de la ap...	26 KB	

11.4. Instalación de dotPeek

1. Descargamos dotPeek desde su pagina oficial.



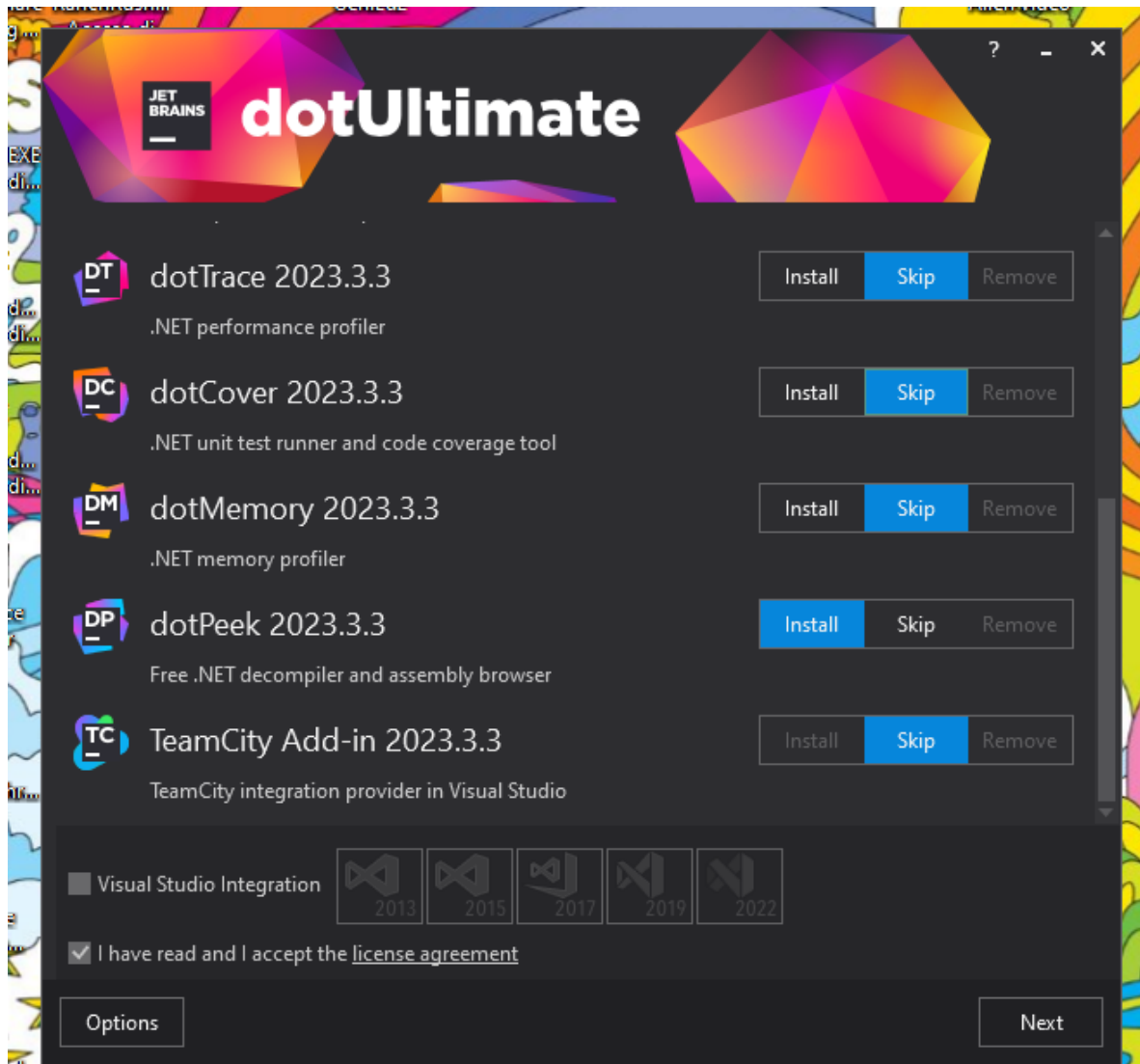
¡Gracias por descargar dotPeek!

La descarga comenzará en breve. Si no es así, utilice el [enlace directo](#).

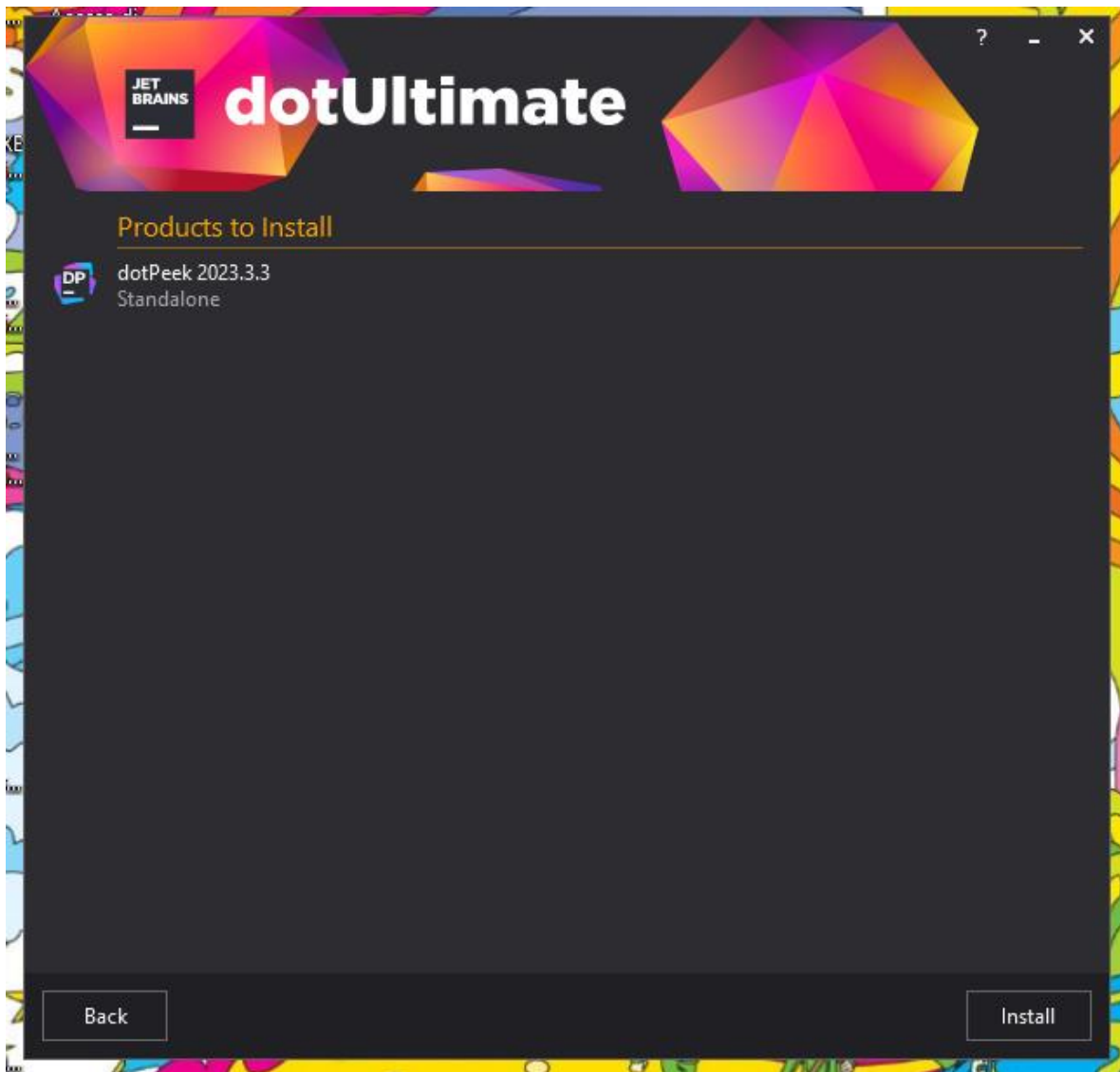
Descargue y verifique el [checksum SHA-256](#) del archivo.

Si tiene una instalación existente de alguna herramienta de dotUltimate, tenga en cuenta las [limitaciones de compatibilidad de dotUltimate](#).

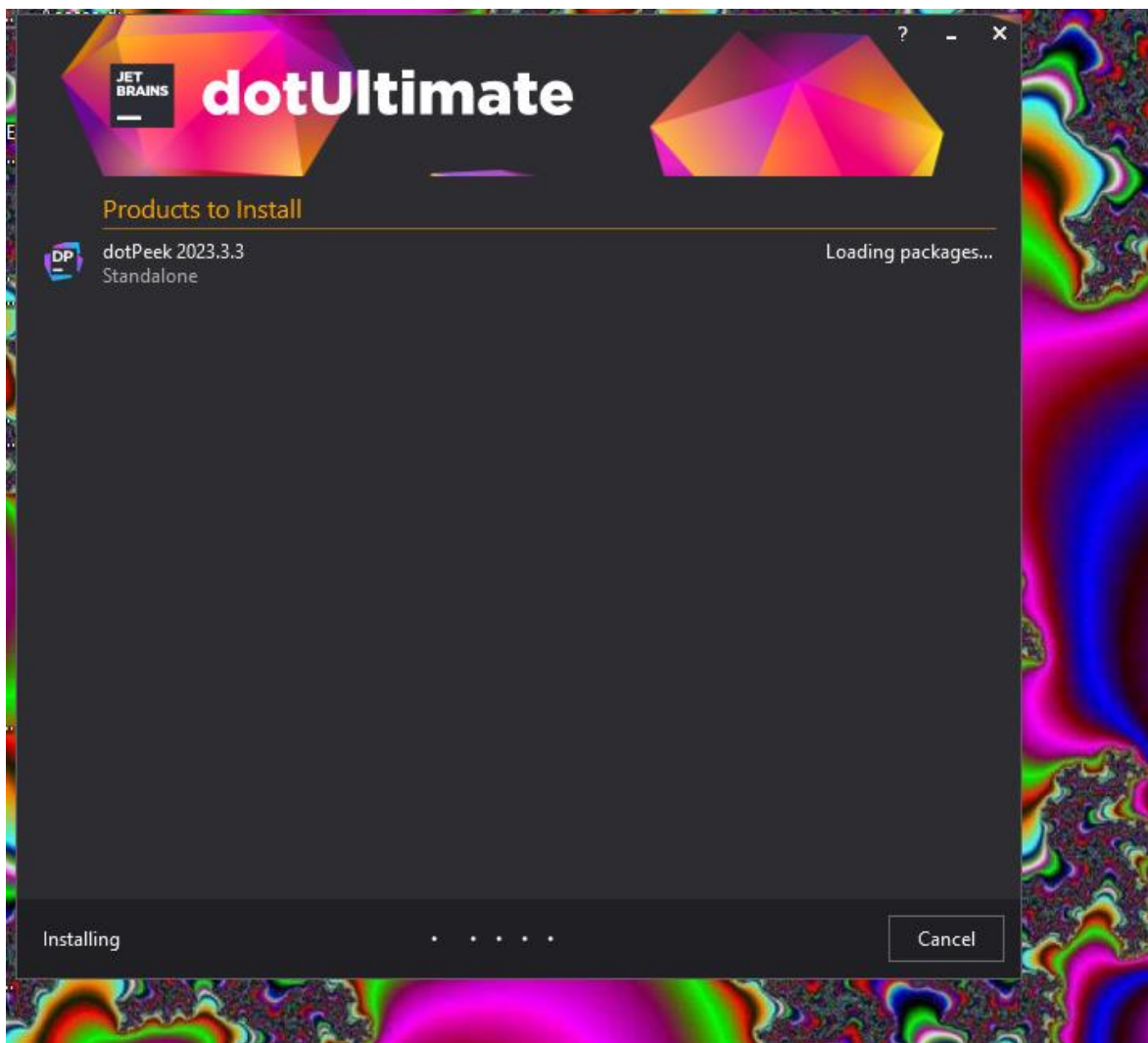
2. Seleccionamos dotPeek y hacemos clic en Next



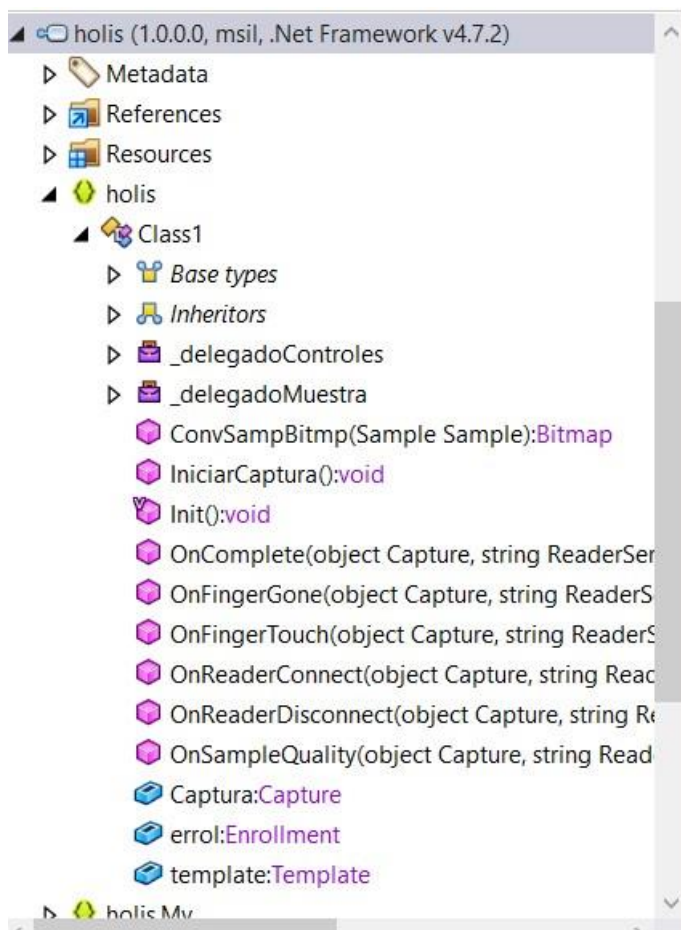
3. Hacemos clic en Install



4. Esperamos a que finalice la instalación.



5. Una vez abierto abrimos nuestra biblioteca DLL en el icono correspondiente.



6. En la parte derecha podremos ver los métodos que hicimos en VisualBasic.

```
{
public class Class1 : DPFP.Capture.EventHandler
{
    public DPFP.Capture.Capture Captura;
    public Enrollment enrol;
    public Template template;

    public virtual void Init()
    {
        try
        {
            this.Captura = new DPFP.Capture.Capture();
            if (this.Captura != null)
            {
                this.Captura.EventHandler = (DPFP.Capture.EventHandler) this;
            }
            else
            {
                int num = (int) MessageBox.Show("No se pudo instanciar la captura");
            }
        }
        catch (Exception ex)
        {
            ProjectData.SetProjectError(ex);
            int num = (int) MessageBox.Show("No se pudo inicializar la captura");
            ProjectData.ClearProjectError();
        }
    }

    public void IniciarCaptura()
    {
    }
}
```

11.5. Instalación de VisualStudioCode

1. Descargamos VisualStudio Code desde la página oficial.



2. Aceptamos el acuerdo y hacemos clic en Siguiente.

Acuerdo de Licencia

Es importante que lea la siguiente información antes de continuar.



Por favor, lea el siguiente acuerdo de licencia. Debe aceptar las cláusulas de este acuerdo antes de continuar con la instalación.

Esta licencia se aplica al producto Visual Studio Code. El código fuente para Visual Studio Code está disponible en <https://github.com/Microsoft/vscode> según el acuerdo de licencia del MIT en <https://github.com/microsoft/vscode/blob/main/LICENSE.txt>. Encontrará información adicional sobre licencias en nuestras preguntas frecuentes en <https://code.visualstudio.com/docs/supporting/faq>.

TÉRMINOS DE LICENCIA DEL SOFTWARE DE MICROSOFT

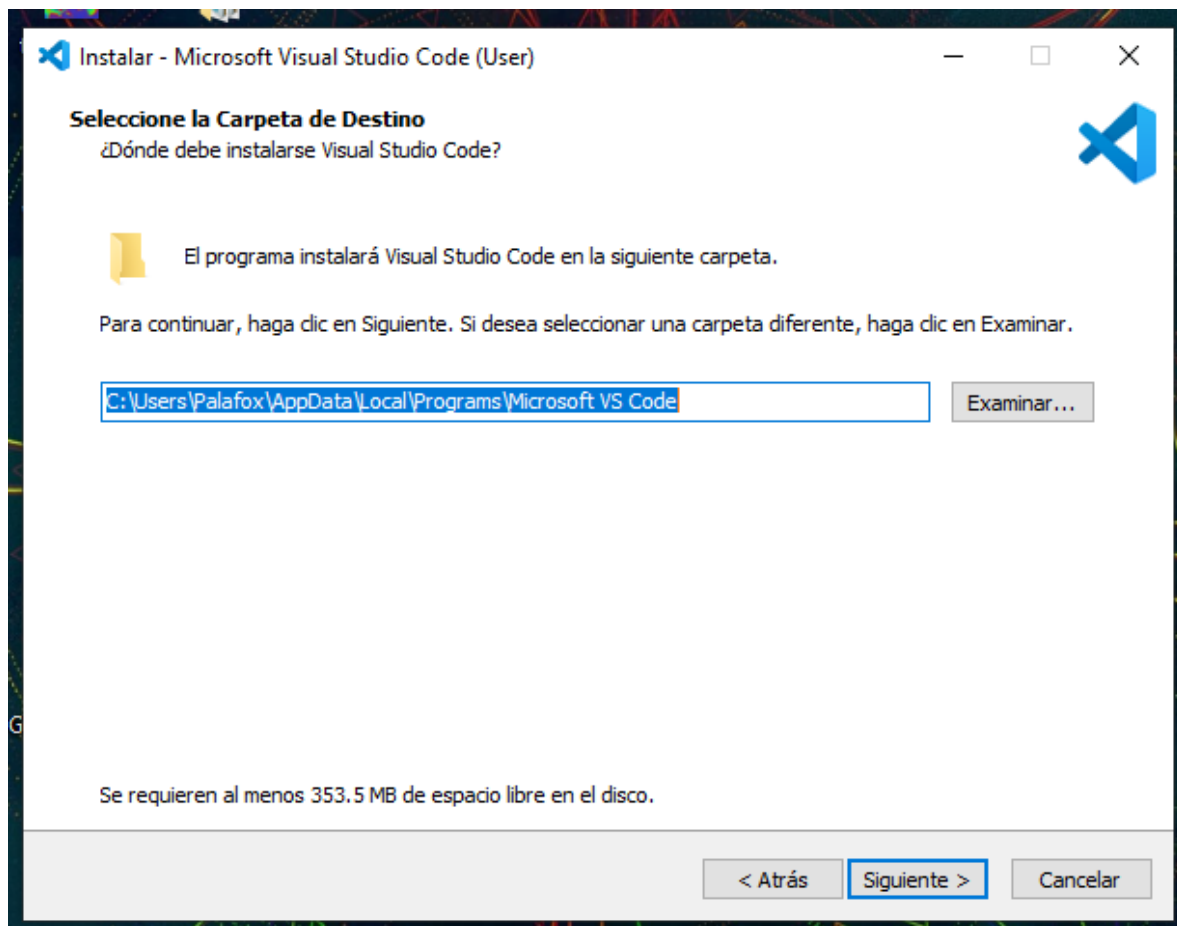
VISUAL STUDIO CODE DE MICROSOFT

- ☒ Acepto el acuerdo
☐ No acepto el acuerdo

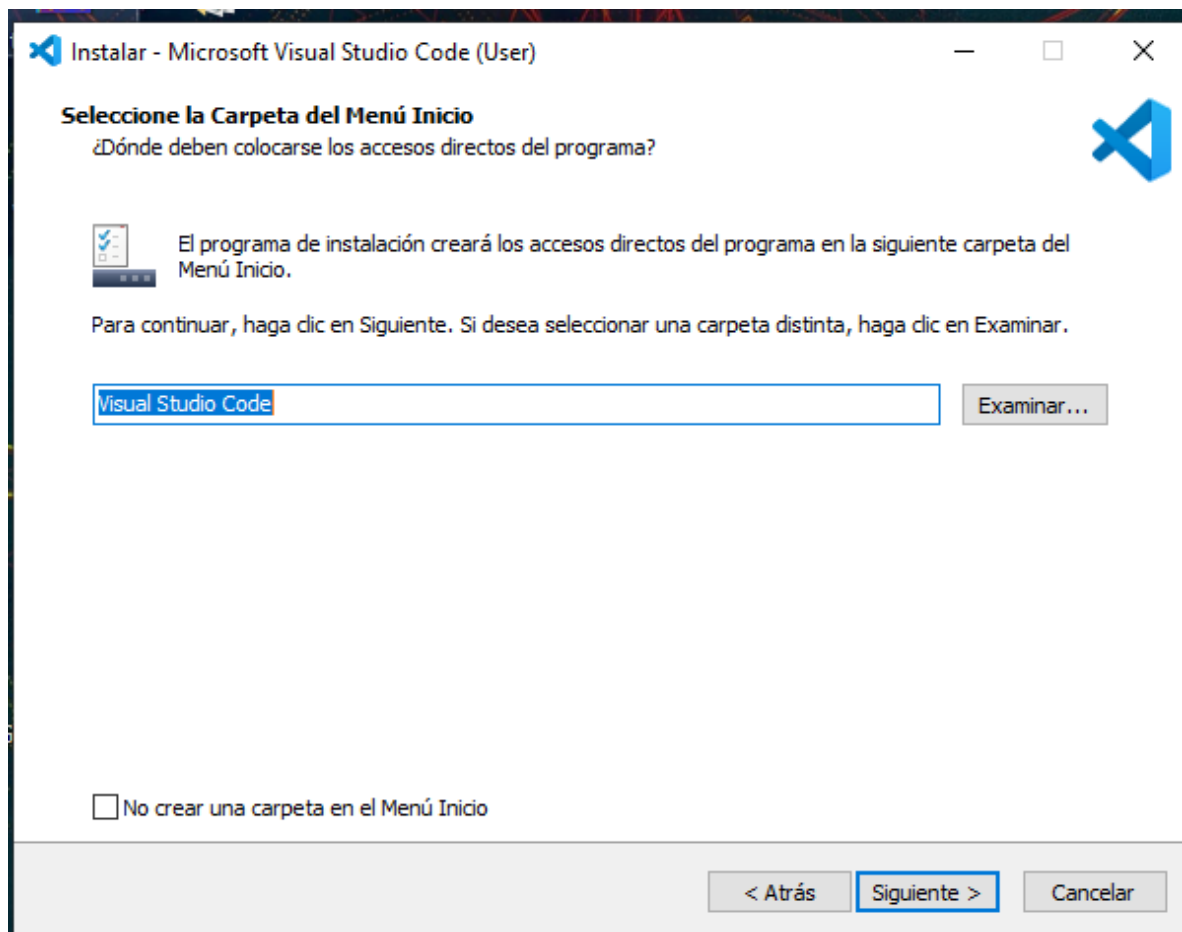
Siguiente >

Cancelar

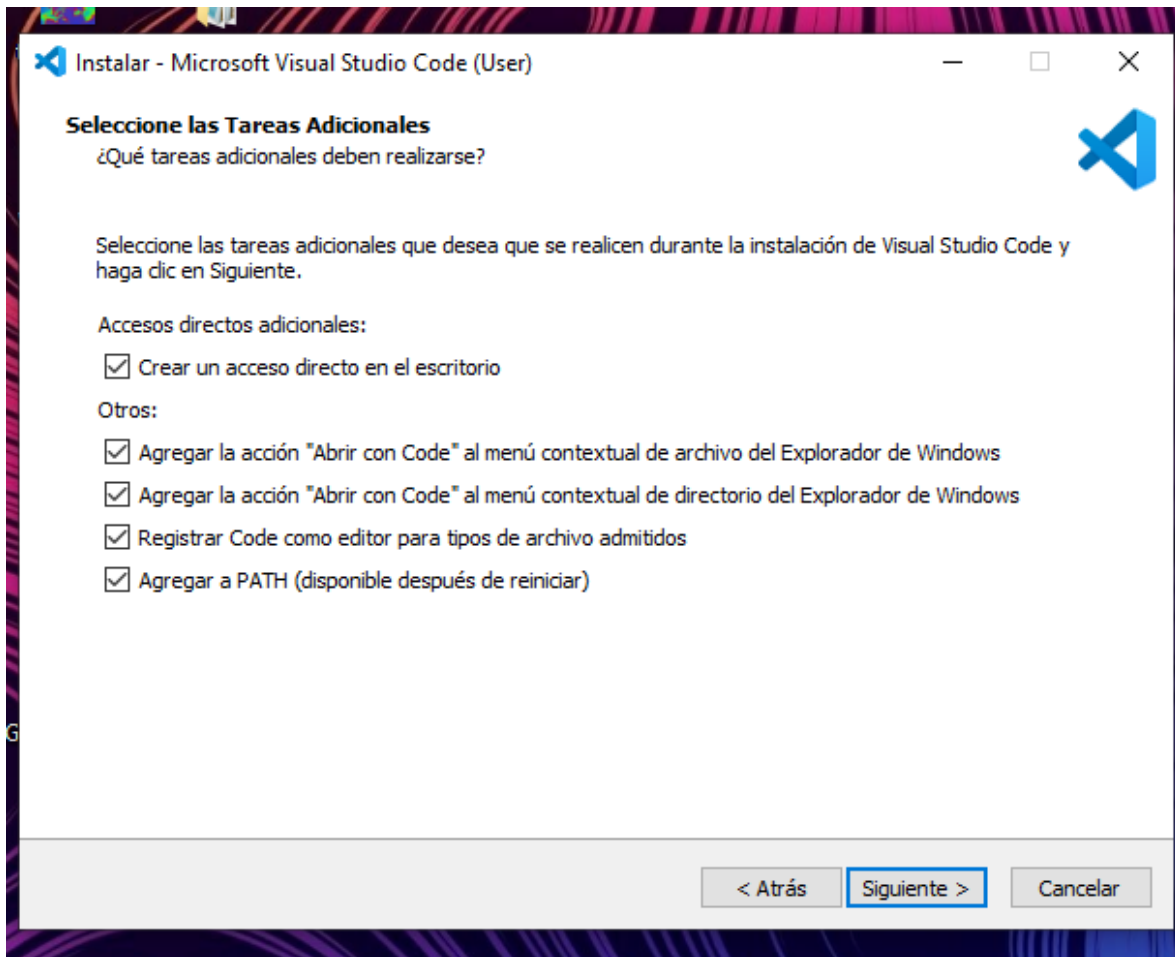
3. Seleccionamos una ubicación y hacemos clic en Siguiente



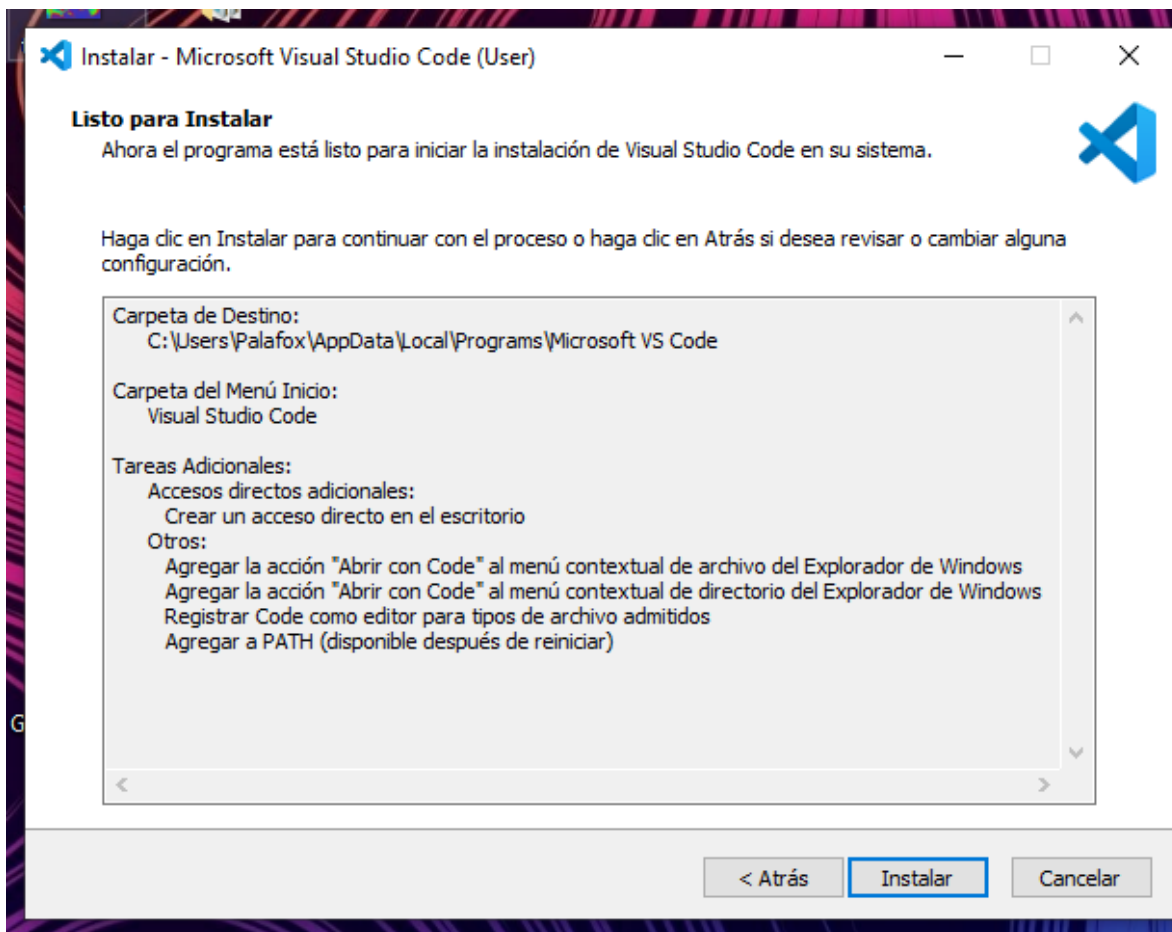
4. Hacemos clic en Siguiente.



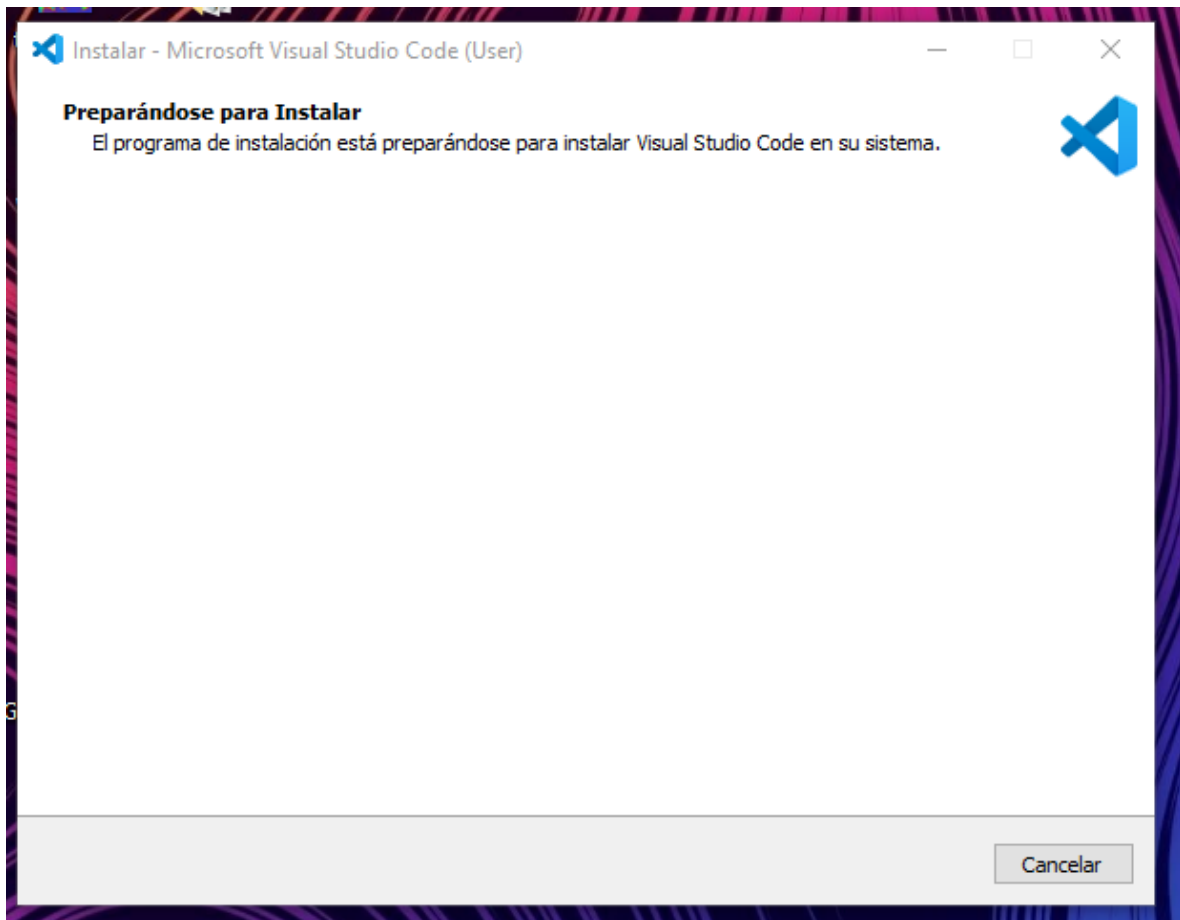
5. Hacemos Clic en siguiente



6. Hacemos clic en Instalar.

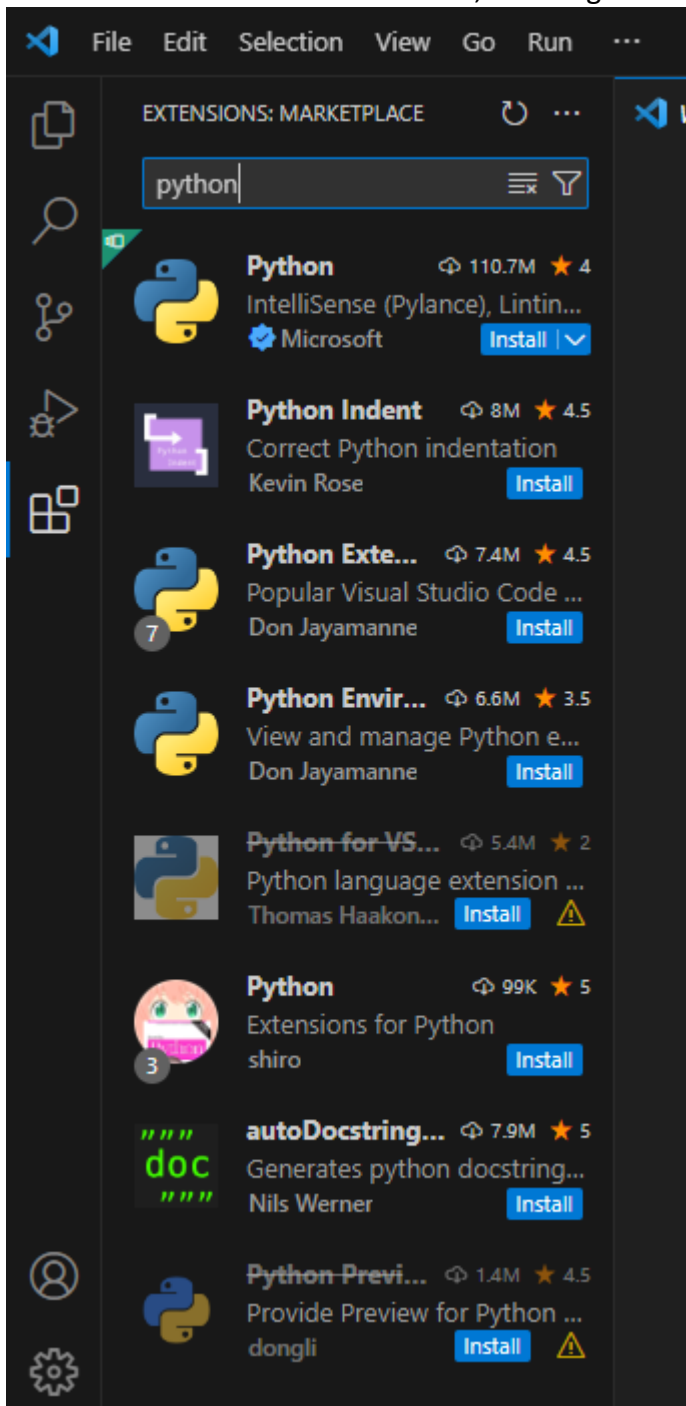


7. Esperamos a que finalice la instalación

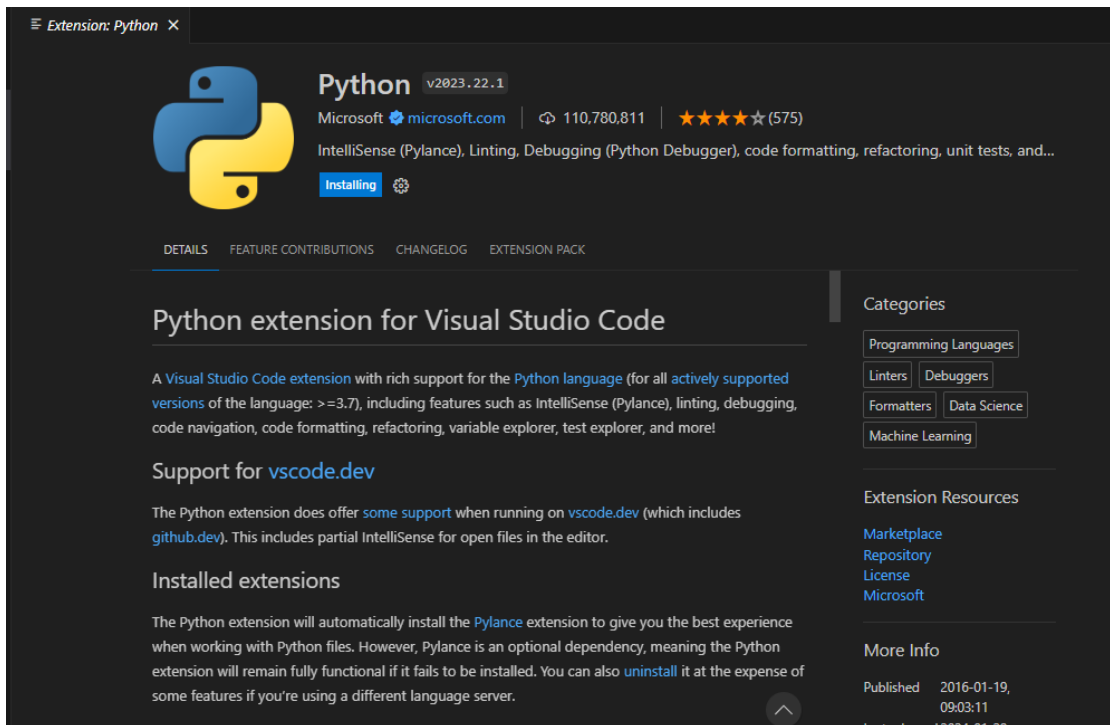


11.6. Instalación de Python

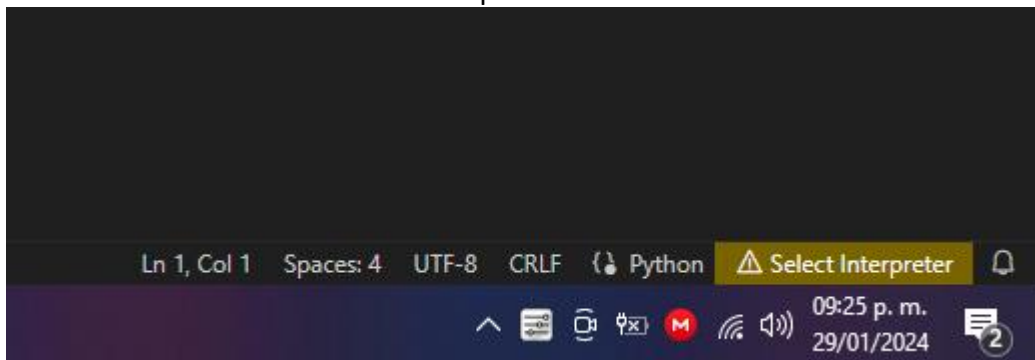
1. Una vez instalado lo abrimos, nos dirigimos a las extensiones e instalamos Python.



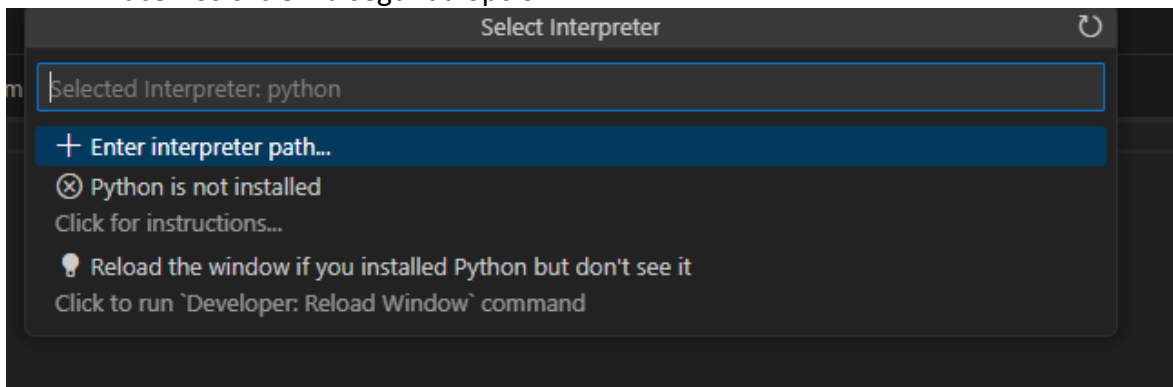
2. Hacemos clic en Install y esperamos



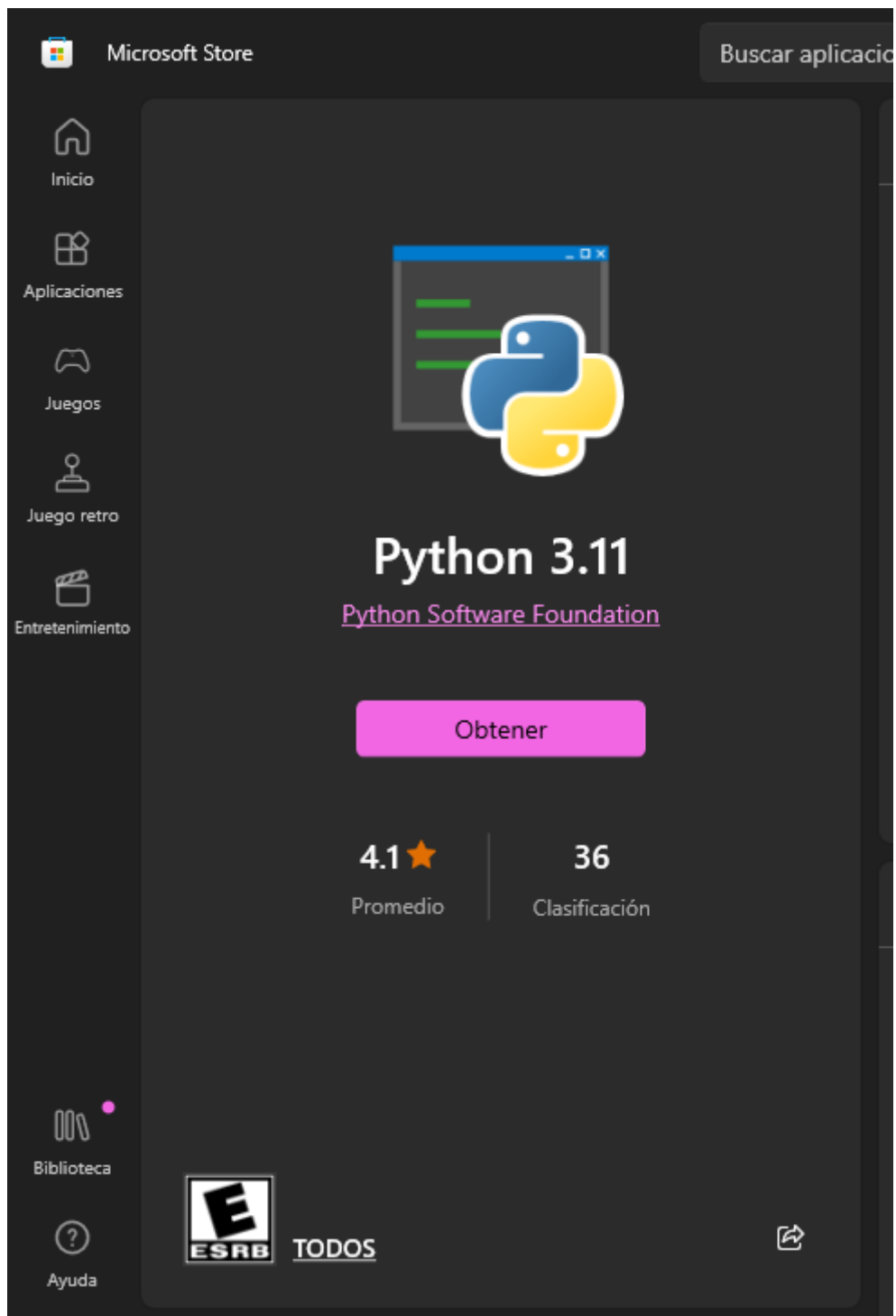
3. Hacemos clic en "Select Interpreter".



4. Hacemos clic en la segunda opción.



5. Hacemos clic en Obtener



6. Instalamos la biblioteca pythonnet

```
23 customIntegerTouch(capture, ReaderSerialNumber)
24

PROBLEMS 11 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\alanh\OneDrive\Documentos\py> pip install pythonnet
```

7. Esperamos a que acabe de instalar.

```
PROBLEMS 11 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\alanh\OneDrive\Documentos\py> pip install pythonnet
Requirement already satisfied: pythonnet in c:\users\alanh\appdata\local\programs\python\python312\lib\site-packages (3.0.3)
Requirement already satisfied: clr-loader<0.3.0,>=0.2.6 in c:\users\alanh\appdata\local\programs\python\python312\lib\site-packages (from p
ythonnet) (0.2.6)
Requirement already satisfied: cffi>=1.13 in c:\users\alanh\appdata\local\programs\python\python312\lib\site-packages (from clr-loader<0.3.
0,>=0.2.6->pythonnet) (1.16.0)
Requirement already satisfied: pycparser in c:\users\alanh\appdata\local\programs\python\python312\lib\site-packages (from cffi>=1.13->clr-
loader<0.3.0,>=0.2.6->pythonnet) (2.21)

[notice] A new release of pip is available: 23.2.1 -> 23.3.2
[notice] To update, run: python.exe -m pip install --upgrade pip
PS C:\Users\alanh\OneDrive\Documentos\py>
```

8. Código de Python:

- Importamos las referencias necesarias.
- Asignamos las referencias a las variables.

```
1 import clr
2 import tkinter as tk
3 from PIL import Image
4
5 clr.AddReference("C:/Users/alanh/OneDrive/Documentos/py/DPFPShrNET.dll")
6 clr.AddReference("C:/Users/alanh/OneDrive/Documentos/py/DPFPDevNET.dll")
7 clr.AddReference("C:/Users/alanh/OneDrive/Documentos/py/DPFPEngNET.dll")
8 clr.AddReference("C:/Users/alanh/OneDrive/Documentos/py/holis.dll")
9
10 from holis import Class1
11 from DPFP import Capture
12 from DPFP import Sample
13
14 ds = Class1()
15 ca = Capture.Capture()
16 sample = Sample()
```

9. Creamos una ventana

```

# Función que se ejecutará al hacer clic en el botón
def mostrar_mensaje():
    #etiqueta.config(text="Hola, has hecho clic en el botón!")
    ds.Init()
    ds.IniciarCaptura()

# Crear una ventana
ventana = tk.Tk()
ventana.title("Mi Primera Ventana")

# Crear un botón
boton = tk.Button(ventana, text="Haz clic", command=mostrar_mensaje)

# Crear una etiqueta
etiqueta = tk.Label(ventana, text="¡Bienvenido a mi ventana!")

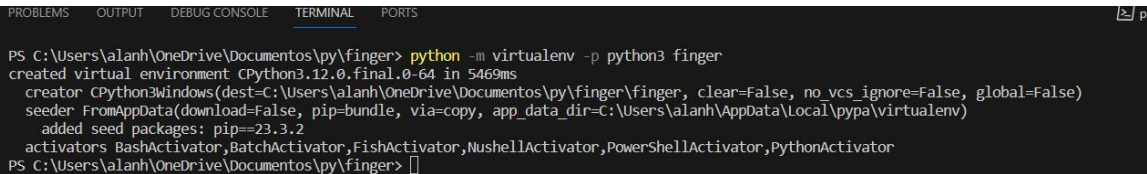
# Posicionar el botón y la etiqueta en la ventana
boton.pack(pady=10)
etiqueta.pack(pady=10)

# Iniciar el bucle principal
ventana.mainloop()

```

11.7. Creación del Proyecto Django

1. Creamos nuestro entorno virtual

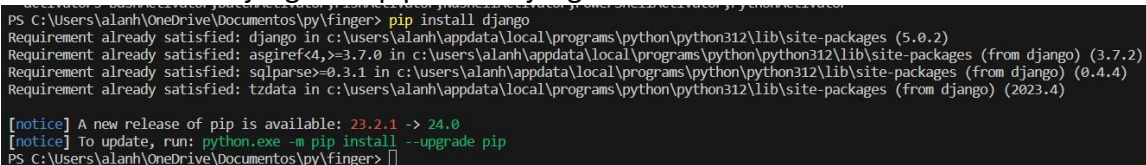


```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\alanh\OneDrive\Documentos\py\finger> python -m virtualenv -p python3 finger
created virtual environment CPython3.12.0.final.0-64 in 5469ms
creator CPythonWindows(dest=C:\Users\alanh\OneDrive\Documentos\py\finger\finger, clear=False, no_vcs_ignore=False, global=False)
seeder FromAppData(download=False, pip=bundle, via=copy, app_data_dir=C:\Users\alanh\AppData\Local\pypa\virtualenv)
added seed packages: pip==23.3.2
activators BashActivator,BatchActivator,FishActivator,NushellActivator,PowerShellActivator,PythonActivator
PS C:\Users\alanh\OneDrive\Documentos\py\finger> 

```

2. Instalamos django con pip install django



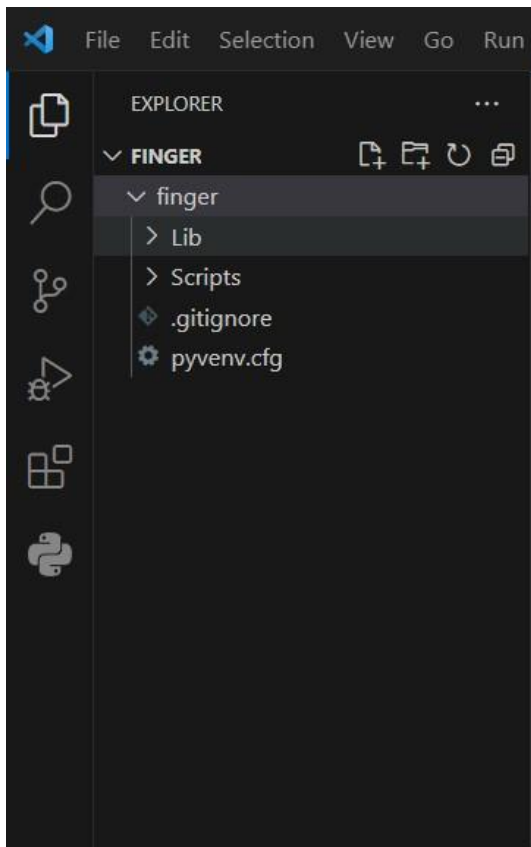
```

PS C:\Users\alanh\OneDrive\Documentos\py\finger> pip install django
Requirement already satisfied: django in c:\users\alanh\appdata\local\programs\python\python312\lib\site-packages (5.0.2)
Requirement already satisfied: asgiref<4,>=3.7.0 in c:\users\alanh\appdata\local\programs\python\python312\lib\site-packages (from django) (3.7.2)
Requirement already satisfied: sqlparse<=0.3.1 in c:\users\alanh\appdata\local\programs\python\python312\lib\site-packages (from django) (0.4.4)
Requirement already satisfied: tzdata in c:\users\alanh\appdata\local\programs\python\python312\lib\site-packages (from django) (2023.4)

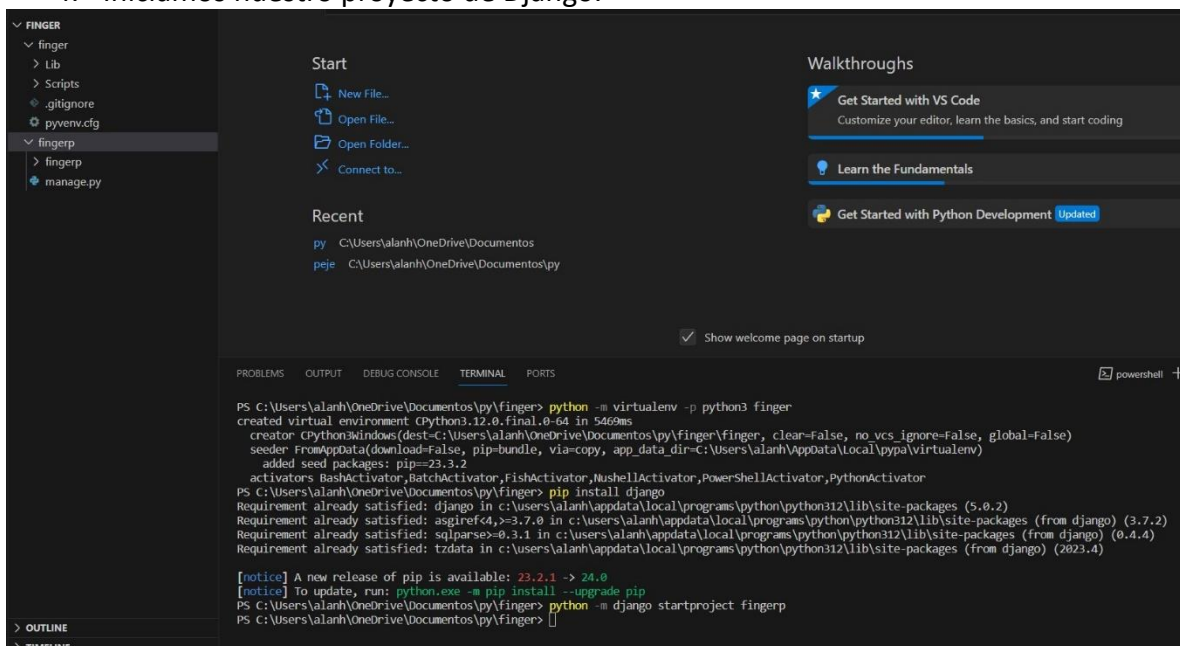
[notice] A new release of pip is available: 23.2.1 -> 24.0
[notice] To update, run: python.exe -m pip install --upgrade pip
PS C:\Users\alanh\OneDrive\Documentos\py\finger> 

```

3. Podremos ver que se ha creado la carpeta del proyecto



4. Iniciamos nuestro proyecto de Django.



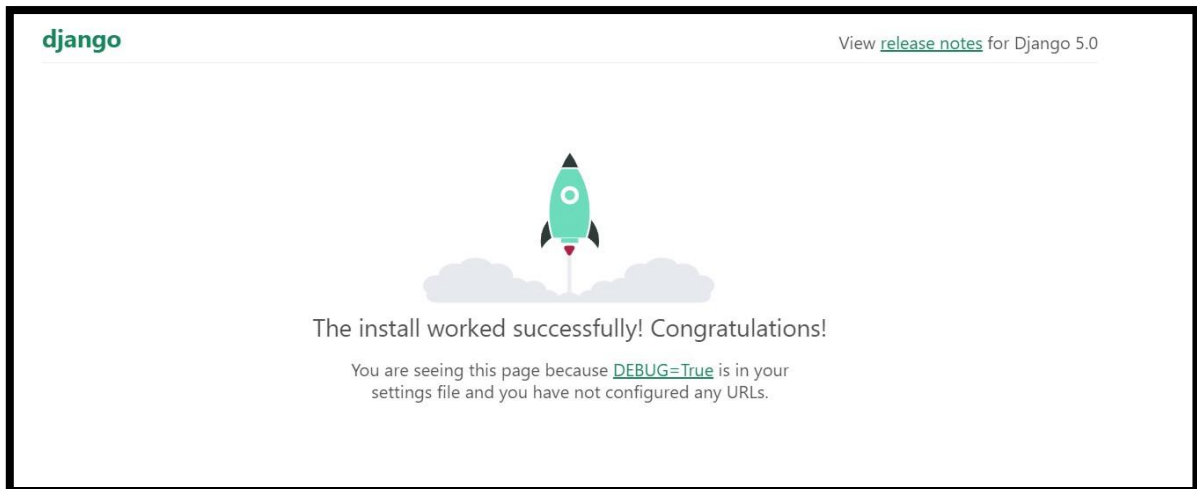
5. Corremos nuestro servidor

```
PS C:\Users\alanh\OneDrive\Documentos\py\finger\finger> python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).

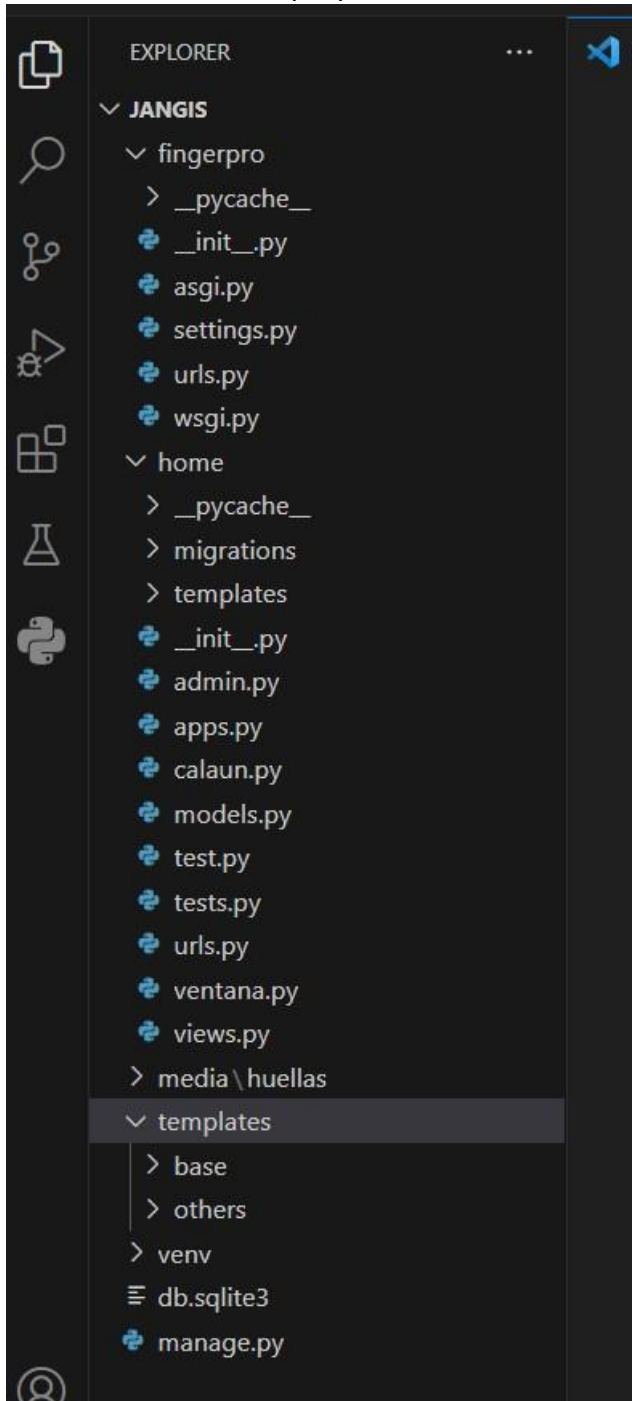
You have 18 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
February 06, 2024 - 21:04:57
Django version 5.0.2, using settings 'fingerp.settings'
Starting development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

6. Podremos ver que se ha configurado exitosamente.



11.8. Código Django

1. Estructura del proyecto.



2. Archivo settings.py: bases de datos (PostgreSQL)

```

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': 'nombre_de_tu_base_de_datos',
        'USER': 'tu_usuario',
        'PASSWORD': 'tu_contraseña',
        'HOST': 'localhost', # O la dirección de tu servidor PostgreSQL
        'PORT': '5432',
    }
}

```

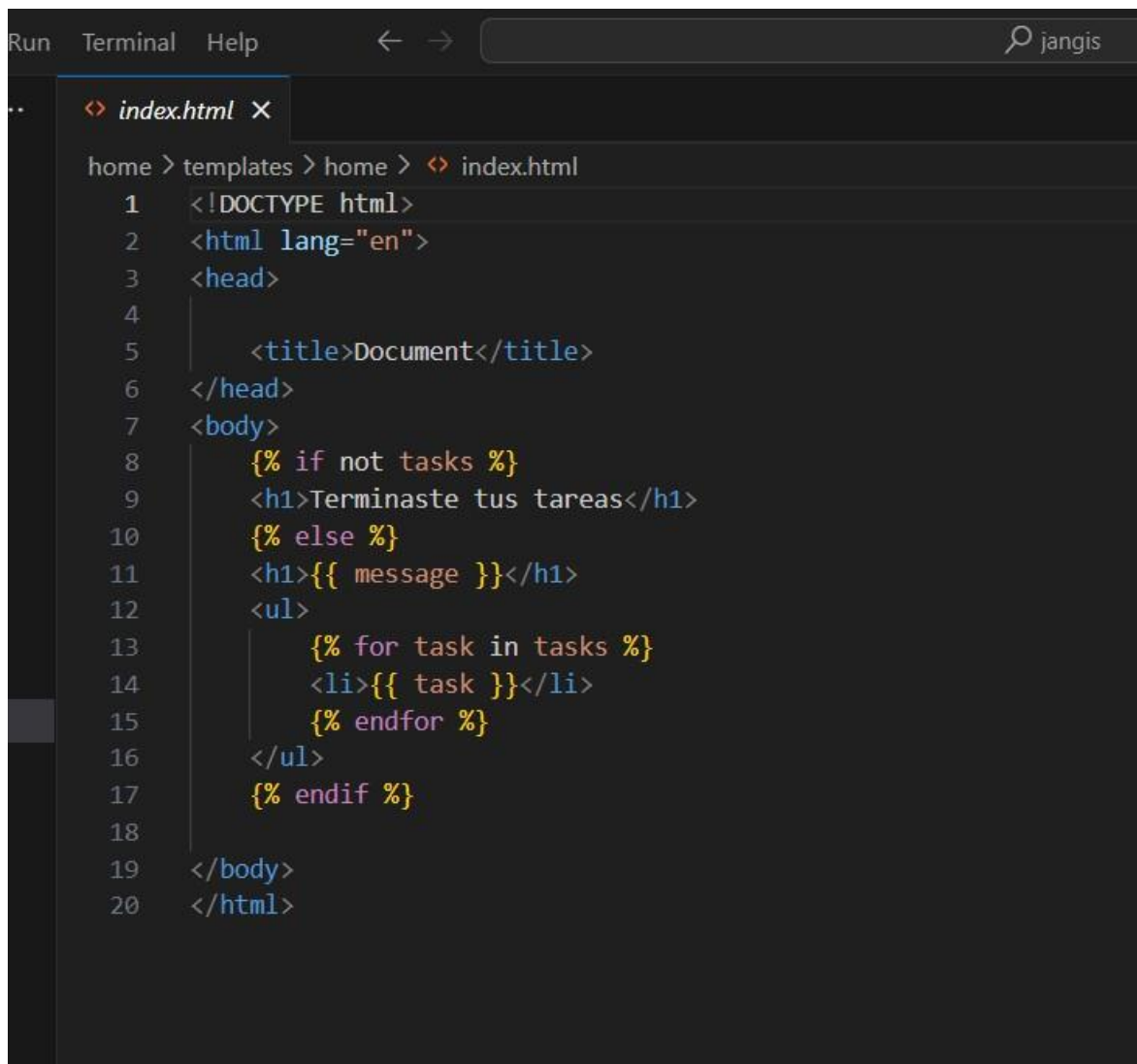
3. Archivo urls.py

```

fingerpro >  urls.py > ...
1  """
2  URL configuration for fingerpro project.
3
4  The `urlpatterns` list routes URLs to views. For more information please see:
5  |   https://docs.djangoproject.com/en/5.0/topics/http/urls/
6  |   Examples:
7  |   Function views
8  |       1. Add an import: from my_app import views
9  |       2. Add a URL to urlpatterns: path('', views.home, name='home')
10 |   Class-based views
11 |       1. Add an import: from other_app.views import Home
12 |       2. Add a URL to urlpatterns: path('', Home.as_view(), name='home')
13 |   Including another URLconf
14 |       1. Import the include() function: from django.urls import include, path
15 |       2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))
16 |   """
17 from django.contrib import admin
18 from django.urls import path, include
19 from django.conf import settings
20 from django.conf.urls.static import static
21
22 urlpatterns = [
23     path('admin/', admin.site.urls),
24     path('', include('home.urls')),
25 ]
26
27
28 if settings.DEBUG:
29     urlpatterns += static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
30

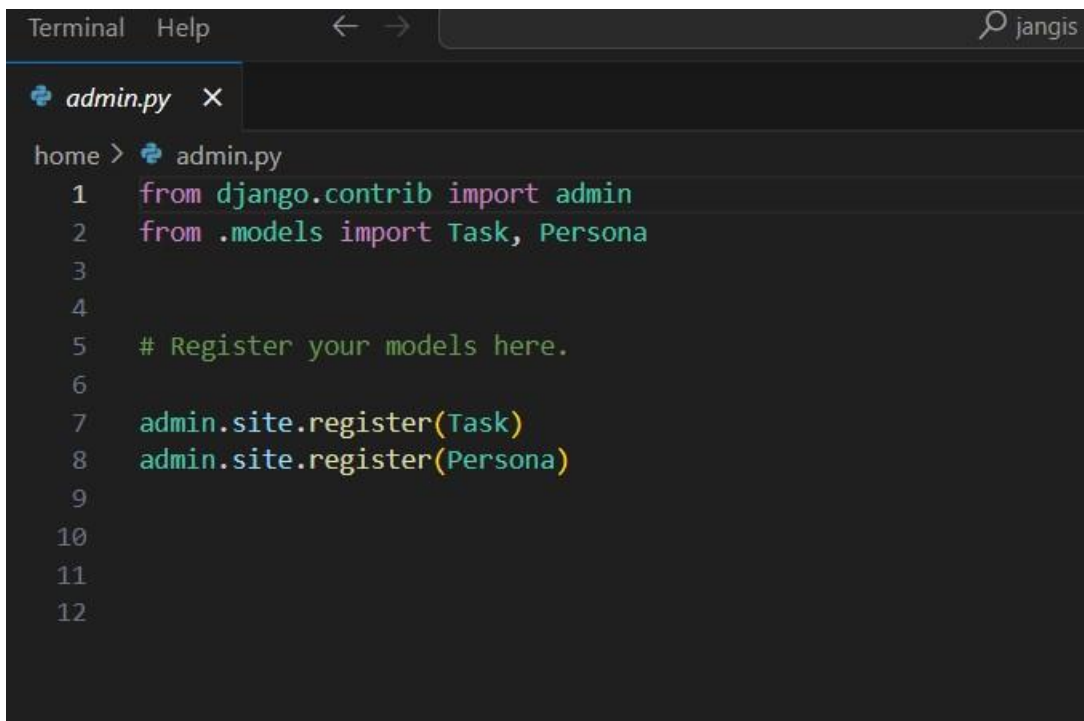
```

4. Archivo index.html



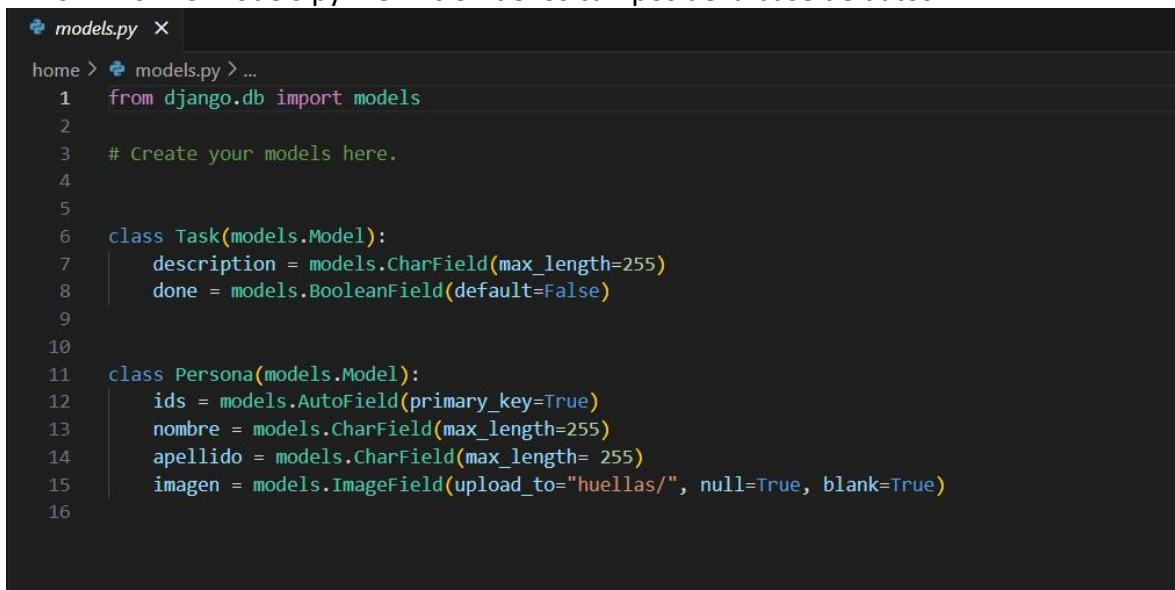
```
Run Terminal Help ← → jangis
index.html X
home > templates > home > index.html
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4
5     <title>Document</title>
6 </head>
7 <body>
8     {% if not tasks %}
9     <h1>Terminaste tus tareas</h1>
10    {% else %}
11    <h1>{{ message }}</h1>
12    <ul>
13        {% for task in tasks %}
14        <li>{{ task }}</li>
15        {% endfor %}
16    </ul>
17    {% endif %}
18
19 </body>
20 </html>
```

5. Archivo admin.py. Configuración del sitio y las pestañas



```
Terminal  Help  < >  jangis
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
12
```

6. Archivo models.py. Definición de los campos de la base de datos.



```
models.py  X
home > models.py > ...
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="huellas/", null=True, blank=True)
16
```

7. Archivo test.py. Aquí es donde se declaran las referencias y las ventanas que aparecerán en pantalla.

```

1
2  import clr
3
4  clr.AddReference('System.IO')
5  clr.AddReference('System.Drawing')
6  clr.AddReference('System.Reflection')
7  clr.AddReference('System.Threading')
8  clr.AddReference('System.Windows.Forms')
9
10 import System
11 import System.IO
12 import System.Drawing
13 import System.Threading
14 import System.Reflection
15 import System.Windows.Forms
16 import System.Windows.Forms
17
18 from System.Windows.Forms import Form
19 from System.Windows.Forms import MessageBox
20 from System.Windows.Forms import PictureBox, PictureBoxSizeMode
21 from System.Drawing import Image
22 from System.Windows.Forms import Label
23
24
25 from System.Threading import Thread, ThreadStart, ApartmentState
26
27
28
29
30
31
32 clr.AddReference("C:/Program Files/DigitalPersona/One Touch SDK/.NET/Bin/DPFPShrNET.dll")
33 clr.AddReference("C:/Program Files/DigitalPersona/One Touch SDK/.NET/Bin/DPFPDevNET.dll")
34 clr.AddReference("C:/Program Files/DigitalPersona/One Touch SDK/.NET/Bin/DPFPEngNET.dll")
35 clr.AddReference("C:/Users/alanh/OneDrive/Documentos/py/Fingerp.dll")
36

```

8. Definimos el constructor y las variables de los elementos que aparecerán en pantalla.

```

62
63 class InteropExplorer(Form):
64
65
66     def __init__(self):
67         super(InteropExplorer, self).__init__()
68
69
70         #Se define el container de los controles
71         self.components = System.ComponentModel.Container()
72
73         print("golAAAAA")
74
75         self.Text = 'Registrar Datos Biometricos'
76
77         self.BackColor = System.Drawing.Color.FromArgb(238, 238, 238)
78
79         self.ClientSize = System.Drawing.Size(600,600)
80
81         caption_height = System.Windows.Forms.SystemInformation.CaptionHeight
82
83
84         self.MinimumSize = System.Drawing.Size(392, (117 + caption_height))
85
86
87         #labels
88
89         self.lbl = System.Windows.Forms.Label()
90         self.lbl.Location = System.Drawing.Point(1, 100)
91         #self.lbl.Text= "Visor de Huellas"
92
93
94         self.perro = Label()
95         #self.perro.Text = "otra cosa"
96         self.perro.Location = System.Drawing.Point(1, 10)
97
98
99         #pictureBox
100
101         self.picture = PictureBox()
102         self.picture.Location = System.Drawing.Point(14, 30)
103         self.picture.SizeMode = PictureBoxSizeMode.Zoom
104
105         #self.picture.Image = Image.FromFile("C:/Users/r2d2999/Documents/queso/salida638421703283650218.png")
106         self.picture.Width = 300
107         self.picture.Height = 300
108
109
110         #Crear boton
111
112         self.LoadFileButton = System.Windows.Forms.Button()
113         self.LoadFileButton.Location = System.Drawing.Point(30, 500)
114         self.LoadFileButton.Size = System.Drawing.Size(256, 54)
115
116         self.LoadFileButton.FlatStyle = System.Windows.Forms.FlatStyle.Flat
117         self.LoadFileButton.Text = "Registrar Datos"
118
119         #definir fuente
120         LoadFileButtonFont = System.Drawing.Font("Roboto", System.Single(12), System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Po
121         self.LoadFileButton.Font = LoadFileButtonFont
122
123         #btn dos
124         self.CloseFileButton = System.Windows.Forms.Button()
125         self.CloseFileButton.Location = System.Drawing.Point(316,500)
126         self.CloseFileButton.Size = System.Drawing.Size(256, 54)
127         self.CloseFileButton.Text = "Cerrar"
128         self.CloseFileButton.Font = LoadFileButtonFont
129         #self.CloseFileButton.FlatStyle = System.Drawing.Font("Roboto", System.Single(12), System.Drawing.FontStyle.Bold, System.Drawing.Gr
130
131         #Se añaden control al form

```

9. Métodos que controlan los botones

```

#Se añaden control al form

self.Controls.Add(self.Lbl)
self.Controls.Add(self.picture)
self.Controls.Add(self.LoadFileButton)
self.Controls.Add(self.CloseFileButton)
self.Controls.Add(self.perro)

#unir eventos al control

self.LoadFileButton.Click += self.btnOnClick

def btnOnClick(self, sender, args):
    #MessageBox.Show("Hola")
    #self.picture.Image = Image.FromFile("C:/Users/r2d2999/Documents/queso/salida638421703283650218.png")
    self.picture.Image = ds.mapas

    #ds.OnComplete(ca, "ReaderSerialNumber", Sample)

def dispose(self):
    self.components.Dispose()
    System.Windows.Forms.Form.Dispose(self)

```

10. Método para invocar la ventana de los elementos.

```

def run(self):
    System.Windows.Forms.Application.Run(self)

def main_form_thread():
    #Inicializar el form
    interop_form = InteropExplorer()

    #impresion en colsona
    print("se creo la isntancia :D")

    #definir el nuevo objeto de forms
    win_form_app = System.Windows.Forms.Application

    #Correr el interop explorer

    win_form_app.Run(interop_form)

if __name__ == '__main__':
    ds.Init()
    ds.IniciarCaptura()
    main_form_thread()

```

11. Archivo urls.py. Aquí definimos las url para poder acceder al sistema diseñado.

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

12. Archivo views.py. Aquí definimos las vistas que aparecerán en la página principal.

```
views.py 4 X
home > views.py > ...
1  from django.shortcuts import render, redirect, reverse
2  from django.http import HttpResponse
3  from .models import Task, Persona
4  import clr
5  import tkinter as tk
6  from PIL import Image, ImageTk
7  from django.core.files import File
8
9
10  clr.AddReference("C:/Program Files/DigitalPersona/One Touch SDK/.NET/Bin/DPFPShrNET.dll")
11  clr.AddReference("C:/Program Files/DigitalPersona/One Touch SDK/.NET/Bin/DPFPDevNET.dll")
12  clr.AddReference("C:/Program Files/DigitalPersona/One Touch SDK/.NET/Bin/DPFPEngNET.dll")
13  clr.AddReference("C:/Users/alanh/OneDrive/Documentos/py/Fingerp.dll")
14
15
16  from Fingerp import Class1
17  from DPFP import Capture
18  from DPFP.Processing import Enrollment
19  from DPFP import Template
20
21
22  ds = Class1()
23  ca = Capture.Capture()
24
25  #dra = Class1.System.Windows.Forms.MessageBox
26
27  # clase de la ventana
28
29
30  class ventanaDOS(tk.Tk):
31
32      def __init__(self):
33          super().__init__()
34
35          self.title("Pruebas")
36          self.geometry("500x500")
37
```

13. Definición de las propiedades de los elementos.

```

38
39 #widgets
40
41 self.lb1 = tk.Label(self, text="Ingresa tus datos")
42 self.lb1.pack(pady=10)
43
44 #nombre
45 self.lb2 = tk.Label(self, text="Nombre")
46 self.lb2.pack(pady=5)
47
48
49 self.ent = tk.Entry(self)
50 self.ent.pack(pady=10)
51
52 #ID
53
54 self.lb3 = tk.Label(self, text="ID NO.")
55 self.lb3.pack(pady=5)
56
57
58 self.ent2 = tk.Entry(self)
59 self.ent2.pack(pady=10)
60
61
62 #botones
63
64 self.button = tk.Button(self, text="Guardar datos", command= self.on_click)
65 self.button.pack(pady=10)
66
67
68 self.buttonPrint = tk.Button(self, text="Agregar Huellas", command=self.on_click_print)
69 self.buttonPrint.pack(pady=10)

```

14. Métodos para mostrar las imágenes y verificar que los campos estén llenos.

```

def checkDataOn(self, campo, campo2):
    if campo and campo2:
        print("tienen algo")

        #dra.MessageBox.Show("sopdjaspodkaposdk")

        return True

    elif not campo:
        print("error, escribe tu nombre")

        return False
    elif not campo2:
        print("error, escriba tu id no.")
        return False
    else:
        print("error, llena ambos campos para continuar")
        return False

def mostrar_imagen(self):
    print("gola")

    #Ruta de la imagen PNG

    ruta_imagen = "C:/Users/alanh/OneDrive/Documentos/py/"+ds.photonames

    # Cargar la imagen con PIL
    imagen_pil = Image.open(ruta_imagen)

    # Convertir la imagen PIL a un formato compatible con Tkinter
    imagen_tk = ImageTk.PhotoImage(imagen_pil)

```

15. Métodos para guardar las imágenes generadas.

```

# Convertir la imagen PIL a un formato compatible con Tkinter
imagen_tk = ImageTk.PhotoImage(imagen_pil)

# Actualizar la etiqueta con la nueva imagen
self.etkImage.config(image=imagen_tk)
self.etkImage.image = imagen_tk # Necesario para evitar que la imagen sea eliminada por el recolector de basura

#update labeltxt
#etiquetatx.config(text=ds.photonames)

def saveData(self, campo, campo2):
    #print("saved")
    neo_persona = Persona.objects.get_or_create(nombre=campo, apellido=campo2)
    #neo_persona.save()

def on_click(self):

    campo = self.ent.get()
    campo2 = self.ent2.get()
    campo3 = 1122
    #print(campo)
    #Verifica campos vacios
    if self.checkDataOn(campo, campo2):
        print("trueeeee")
        #ds.Init()
        #ds.IniciarCaptura()

        #self.saveData(campo, campo2)
    else:
        print("falseeeee")

```

16. Método para mostrar las imágenes.

```

0
1
2     campo = self.ent.get()
3     campo2 = self.ent2.get()
4     campo3 = 1122
5     #print(campo)
6     #Verifica campos vacios
7     if self.checkDataOn(campo, campo2):
8         print("trueeeee")
9         #ds.Init()
10        #ds.IniciarCaptura()
11
12        #self.saveData(campo, campo2)
13    else:
14        print("falseeeee")
15
16    def on_click_print(self):
17        print("imagen show")
18        self.mostrar_imagen()
19
20        #ruta_save = "C:/Users/r2d2999/Documents/queso/salida638421703399418377.png"
21        ruta_save = "C:/Users/alanh/OneDrive/Documentos/py/" + ds.photonames
22
23        instancia = Persona(nombre = "juana", apellido = "Quesoos")
24
25        with open(ruta_save, 'rb') as archivo:
26            image_file = File(archivo)
27            instancia.imagen.save('imagen.png', image_file)
28
29
30        instancia.save()
31
32
33

```

17. Modelo html

```
views.py 4  master.html x
templates > base > master.html
6  <title>
7      {% block title %}
8      {% endblock %}
9  </title>
10 </head>
11 <body>
12     <nav class="navbar navbar-expand-lg bg-body-tertiary">
13         <div class="container-fluid">
14             <a class="navbar-brand" href="#">Navbar</a>
15             <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls=
16                 <span class="navbar-toggler-icon"></span>
17             </button>
18             <div class="collapse navbar-collapse" id="navbarSupportedContent">
19                 <ul class="navbar-nav me-auto mb-2 mb-lg-0">
20                     <li class="nav-item">
21                         <a class="nav-link active" aria-current="page" href="#">Home</a>
22                     </li>
23                     <li class="nav-item">
24                         <a class="nav-link" href="#">Link</a>
25                     </li>
26                     <li class="nav-item dropdown">
27                         <a class="nav-link dropdown-toggle" href="#" role="button" data-bs-toggle="dropdown" aria-expanded="false">
28                             Dropdown
29                         </a>
30                         <ul class="dropdown-menu">
31                             <li><a class="dropdown-item" href="#">Action</a></li>
32                             <li><a class="dropdown-item" href="#">Another action</a></li>
33                             <li><hr class="dropdown-divider"></li>
34                             <li><a class="dropdown-item" href="#">Something else here</a></li>
35                         </ul>
36                     </li>
37                     <li class="nav-item">
38                         <a class="nav-link disabled" aria-disabled="true">Disabled</a>
39                     </li>
40                 </ul>
41                 <form class="d-flex" role="search">
42                     <input class="form-control me-2" type="search" placeholder="Search" aria-label="Search">
```

12. Resultados obtenidos

1. Interfaz para solicitar huellas y datos.

Form1

Pasa el dedo 4 veces

Nombre

Guardar

Find

2. Faltan 3 intentos:

Form1



Pasa el dedo 3 veces

Nombre

Guardar

Find

3. Faltan 2 intentos:

Form1



Pasa el dedo 2 veces

Nombre

Guardar

Find

4. Falta un intento

Form1



Pasa el dedo 1 veces

Nombre

Guardar

Find

5. Una vez colocada 4 veces la huella se habilitará la casilla de nombre, donde podremos agregar nuestros datos.

Form1



Pasa el dedo 0 veces

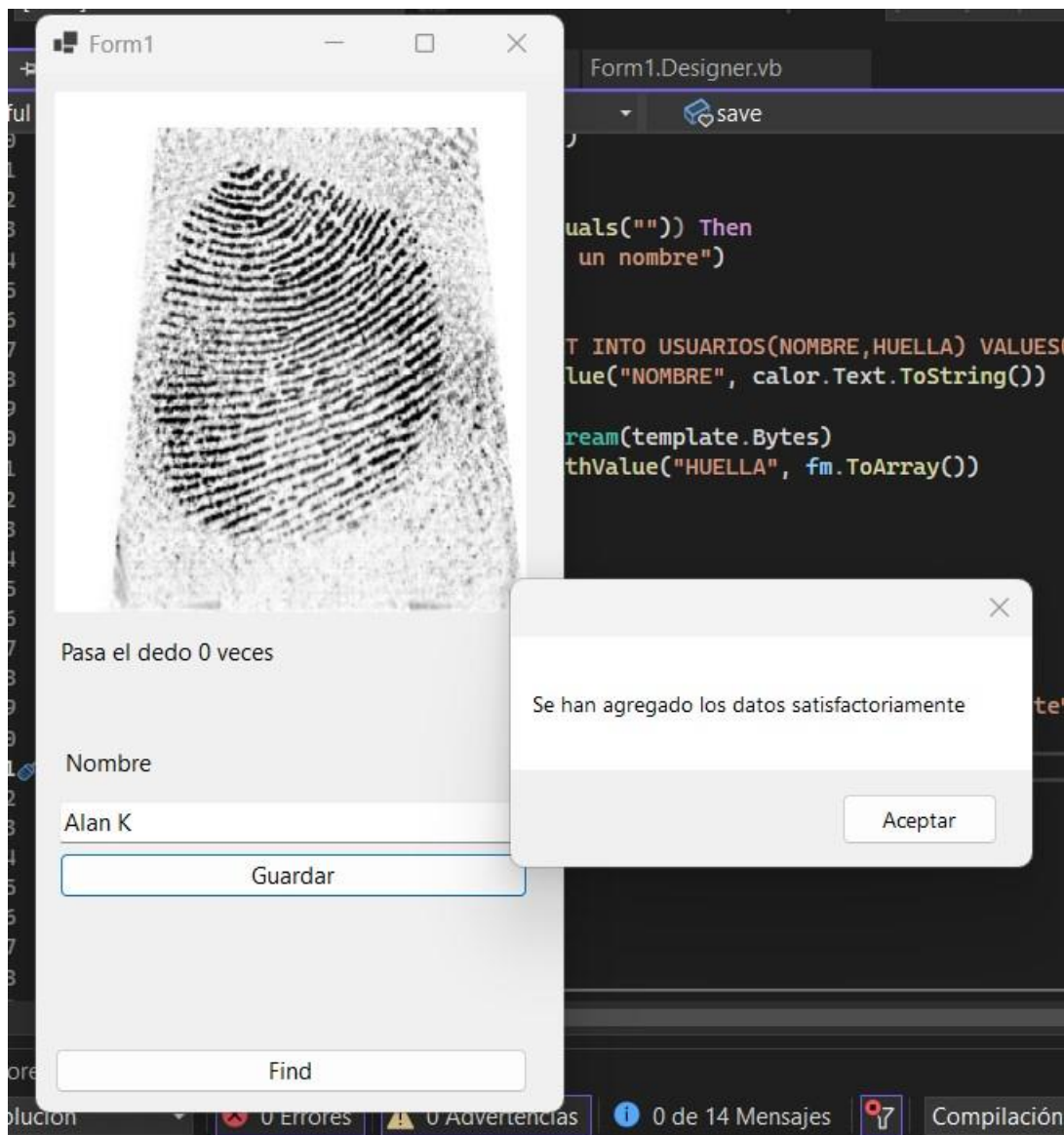
Nombre

Alan K

Guardar

Find

6. Al hacer clic en guardar nos mostrara un mensaje de hecho.



7. Si verificamos en la base de datos veremos que se ha guardado la información junto con las huellas

phpMyAdmin

Recent Favorites

New fingerprint

New usuarios

Columns

Indexes

information_schema

mysql

performance_schema

phpmyadmin

test

Server: 127.0.0.1 » Database: fingerprint » Table: usuarios

Browse Structure SQL Search Insert

Showing rows 0 - 3 (4 total, Query took 0.0006 seconds.)

`SELECT * FROM `usuarios``

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search

Extra options

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

8. Interfaz de DJANGO

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

Registrar Datos Biometricos

Ingresa tu nombre

Javier

Ingresa tu apellido

hermes



Registrar Datos

Cerrar

Object Explorer

Search objects (Alt+Shift+S)

Database: personas/postgres@PostgreSQL 16

Query: `select * from home_people_register;`

Data Output

id [PK] bigint	nombre character varying (100)	apellido character varying (100)	imagen character varying (100)	imgjson jsonb
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	""

9. Imágenes guardadas

media > huellas > imagen_SuoITga.png

media > huellas > imagen_SuoITga.png



13. Conclusiones

En conclusión, el proyecto de Análisis y Experimentación para Extracción y Selección de Características para Huellas Dactilares ha sido un paso significativo hacia la mejora de los sistemas de identificación biométrica basados en este método. A través del uso de tecnologías como Python y Visual Basic, junto con herramientas como Visual Studio y Visual Studio Code, hemos logrado avanzar en la comprensión y aplicación de técnicas de procesamiento de imágenes y selección de características en el contexto de las huellas dactilares.

Durante el desarrollo del proyecto, hemos identificado varios puntos clave, como la combinación de lenguajes de programación como Python y Visual Basic ha permitido una implementación efectiva y eficiente de los algoritmos necesarios para la extracción y selección de características de las huellas dactilares. El uso de Visual Studio y Visual Studio Code como entornos de desarrollo ha proporcionado herramientas poderosas para la escritura, depuración y optimización del código, lo que ha contribuido a la eficacia y calidad del software desarrollado. La integración de bibliotecas DLL y otras herramientas de software ha enriquecido el proyecto, permitiendo el acceso a funcionalidades adicionales y mejorando la interoperabilidad con otros sistemas y tecnologías. Se han realizado experimentos exhaustivos para evaluar la efectividad y eficiencia de los algoritmos implementados, utilizando conjuntos de datos representativos y métricas de evaluación adecuadas.

Los resultados obtenidos durante la experimentación han demostrado mejoras significativas en la precisión y confiabilidad de los sistemas de identificación biométrica basados en huellas dactilares, lo que sugiere un potencial considerable para su aplicación en una variedad de campos, desde la seguridad hasta la gestión de identidades. En resumen, el proyecto ha sido un éxito en términos de avance científico y tecnológico en el campo de la identificación biométrica. Las lecciones aprendidas y los resultados obtenidos proporcionan una base sólida para futuras investigaciones y desarrollos en esta área, con el objetivo final de mejorar la seguridad y eficiencia de los sistemas biométricos en todo el mundo.

14. Competencias Desarrolladas

- ✚ Adquiere habilidades en el procesamiento de imágenes para manipular y analizar huellas dactilares, incluyendo técnicas de filtrado, segmentación y mejora de la calidad de imagen.
- ✚ Comprende de los principios y conceptos fundamentales de la biometría, especialmente en relación con la identificación y verificación de huellas dactilares.
- ✚ Domina la programación en Python y Visual Basic para implementar algoritmos de extracción y selección de características, así como para desarrollar interfaces de usuario y sistemas de gestión de datos.
- ✚ Desarrolla de software mediante el uso de herramientas como Visual Studio y Visual Studio Code para escribir, depurar y optimizar el código.
- ✚ Aprende sobre los aspectos técnicos y prácticos relacionados con el desarrollo de aplicaciones biométricas, incluyendo la integración de lectores biométricos, la gestión de datos biométricos y la seguridad de la información.
- ✚ Desarrolla habilidades en la evaluación y experimentación de algoritmos y técnicas de extracción y selección de características, utilizando conjuntos de datos reales o simulados y métricas de evaluación adecuadas.
- ✚ Entiende los principios de interoperabilidad de software y la integración de bibliotecas DLL y otros componentes en aplicaciones desarrolladas en Python y Visual Basic.
- ✚ Gestiona de proyectos de desarrollo de software, incluyendo la definición de objetivos, la asignación de recursos y el seguimiento del progreso del proyecto.
- ✚ Analiza los datos para interpretar y extraer conclusiones a partir de los resultados de experimentos y evaluaciones realizadas durante el proyecto.
- ✚ Desarrolla habilidades de comunicación efectiva y trabajo en equipo para colaborar con otros miembros del equipo, compartir conocimientos y resultados, y presentar hallazgos de manera clara y concisa.

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>