



MANUAL TÉCNICO

“PIPESOUND”

ELABORADO POR

JAIME CABEZAS SEGURA
NATASHA CALDERÓN ROJAS
SEBASTIÁN LÓPEZ VILLAVICENCIO
JOSÉ ANDRÉS RAMÍREZ CASTILLO

SAN JOSÉ
2025

RESUMEN

El presente proyecto tiene como objetivo desarrollar un pipeline automatizado para el procesamiento de archivos de audio capturados con placas AudioMoth y subidos a la plataforma Arbimon.org. Actualmente, el flujo de trabajo requiere la descarga manual de los archivos desde Arbimon, su conversión al formato WAV, y el procesamiento mediante R, utilizando una librería desarrollada por Oscar Ramírez. Este enfoque manual resulta ineficiente y propenso a errores, por lo que se propone la implementación de un pipeline en Python que automatice tareas clave como la descarga de audios, su transformación al formato requerido y la gestión de la información procesada. Además, Python ofrece acceso a una variedad de librerías especializadas en análisis de audio que proporcionan indicadores más completos, superando algunas limitaciones de R.

Palabras clave: keyword1, keyword2

Índice general

1. Introducción	1
1.1. Sobre este manual	1
1.2. Descripción y Alcance del proyecto	1
1.2.1. Antecedentes	1
1.2.2. Objetivos	2
1.2.3. Contacto	2
2. Especificación de funcionalidad	3
2.1. Descripción del producto	3
2.1.1. Historias de usuario	3
3. Arquitectura del sistema	8
3.1. Diseño general del sistema	8
3.1.1. Diagrama de clases	8
4. Instrucciones de Instalación	9
4.1. Entorno de Desarrollo	9
4.1.1. Dependencias	9
4.1.2. Código fuente	9
4.1.3. Despliegue en entorno de pruebas	12
4.2. Entorno de Producción	13
4.2.1. Dependencias	13
4.2.2. Instrucciones de instalación	13
4.3. Requerimientos del sistema	13
4.3.1. Requerimientos mínimos	13
4.3.2. Requerimientos recomendados	13
5. Conclusiones y trabajo futuro	14
5.1. Conclusiones	14
5.2. Problemáticas y limitaciones	14
5.3. Trabajo futuro	14
Referencias bibliográficas	15

Índice de figuras

3.1. Diagrama de clases	8
4.1. Python version	10
4.2. Python Installer	10
4.3. Python Installer	11
4.4. Python Installer	11
4.5. Verificación	12
4.6. Instalación Bibliotecas	12

Capítulo 1

Introducción

1.1. Sobre este manual

Este manual consiste de una descripción del proyecto PIPESOUND y su alcance. Una descripción de las funcionalidades del sistema y como está construido.

El propósito de este manual es ayudar al usuario a comprender la estructura del mismo.

1.2. Descripción y Alcance del proyecto

1.2.1. Antecedentes

El proyecto PIPESOUND fue creado junto con el Laboratorio Investigación e Innovación Tecnológica y la Universidad Estatal a Distancia. LIIT desarrolla iniciativas de investigación e innovación tecnológica abierta que respondan a las necesidades de la sociedad. Promueve el uso de herramientas de bajo costo y alto impacto mediante un enfoque sistemático aplicado en ciencias ambientales, educativas y computacionales (de Investigación e Innovación Tecnológica (LIIT), s.f.)

El principal problema del sistema radica en la ausencia de un pipeline bien definido para el procesamiento de las grabaciones, todas las grabaciones deben pasar por un mismo conjunto de filtros sin la posibilidad de aplicar configuraciones distintas según las necesidades específicas de cada archivo.

Esta falta de flexibilidad significa que el usuario no puede seleccionar un subconjunto de grabaciones y procesarlo con filtros específicos antes de aplicar otro conjunto de filtros a un grupo diferente, esto genera una ineficiencia en el procesamiento, ya que cualquier ajuste en los filtros requiere repetir todo el procedimiento para cada variación, en lugar de permitir configuraciones diferenciadas dentro de un mismo flujo de trabajo, incrementando así el tiempo que requiere.

El proceso manual de descarga y procesamiento de archivos en la plataforma Arbimon.org es propenso a errores y resulta ineficiente en términos de tiempo y esfuerzo. La automatización de la descarga de archivos mediante el SDK proporcionado por Arbimon facilitará este procedimiento para el usuario, permitiéndole realizar sus tareas de manera más adecuada y reduciendo su carga de trabajo.

Además, la conversión automática de archivos FLAC a WAV agilizará el proceso, asegurando que una mayor cantidad de archivos estén listos para su análisis. Los usuarios podrán seleccionar, de una lista predefinida, los índices o métricas a utilizar en el procesamiento, así como modificar sus parámetros según sus necesidades.

Este enfoque no solo optimiza el flujo de trabajo y reduce la intervención manual, sino que también permite la incorporación de nuevos parámetros y funcionalidades en futuras versiones, garantizando la escalabilidad y adaptabilidad del software. En general, se espera que la implementación del pipeline automatizado facilite el estudio y mejore la capacidad de análisis de los datos de audio recolectados en los proyectos de paisaje acústico realizados por la UNED.

1.2.2. Objetivos

■ Objetivo General

- Desarrollar un pipeline automatizado y configurable que descargue las grabaciones de audio, realice la conversión de las grabaciones de audio a formato WAV, genere un archivo csv con los índices que el usuario seleccione, muestre gráficos de violin de los índices seleccionados e identifique en una grabación de audio o un grupo de grabaciones cuales especies de aves existen.

■ Objetivos Específicos

- Desarrollar una aplicación que permita la selección de múltiples grabaciones de audio en conjunto con el grupo de indicadores conformado por Acoustic Complexity Index, Acoustic Diversity Index, Acoustic Evenness Index, Bioacoustic Index, Normalized Difference sound, Spectral entropy, Temporal entropy, Wave signal to noise ratio y Number of peaks generando como producto un archivo csv con los datos recopilados.
- Desarrollar las funcionalidades de descarga automática de las grabaciones de audio desde el sitio Arbimon y la conversión de los archivos a formato WAV y generación de gráficos de violin utilizando los índices seleccionados.
- Desarrollar la funcionalidad que analiza una grabación o un grupo de grabaciones que determine cuales especies de aves se encuentran en este sitio.

1.2.3. Contacto

Para consultas y soporte, contacte a:

- jacabezas@estudiantec.cr
- selopez@estudiantec.cr
- nacalderon@estudiantec.cr
- jos_ramirez@estudiantec.cr

Capítulo 2

Especificación de funcionalidad

2.1. Descripción del producto

PIPESOUND es una aplicación de escritorio, esta permitirá seleccionar una carpeta que contiene archivos de audio .wav y este genera un archivo csv con el análisis de estos audios. El usuario puede seleccionar distintos índices para analizar. La aplicación además tiene en caso de fallas como que se cierre la aplicación, esta puede recuperar su progreso.

2.1.1. Historias de usuario

MVP 1

Código	Descripción	Criterios de Calidad
RFXXX	Título de la historia: Descripción de la funcionalidad que representa la historia, lo más detallado posible y con énfasis en la implementación de funcionalidad.	Descripción de cómo el usuario puede verifica que la funcionalidad está completa de forma satisfactoria.
RF001	Prototipo de la aplicación: La aplicación debe contar con un prototipo funcional accesible para revisión y pruebas por parte del investigador. Este prototipo debe mostrar la estructura general de navegación, menús y funciones principales.	El investigador puede acceder al prototipo desde un enlace o ejecutable y navegar por sus funcionalidades básicas.
RF002	Visualización de indicadores: La interfaz debe presentar una lista completa de los indicadores disponibles para que el investigador los pueda visualizar y seleccionar.	El investigador puede ver todos los indicadores listados en pantalla y seleccionar uno o varios para análisis.
RF003	Selección del indicador Acoustic Complexity: La aplicación debe permitir seleccionar el indicador "Acoustic Complexity" para su procesamiento.	Al seleccionar este indicador, el sistema debe confirmar su selección e incluirlo en el análisis.
RF004	Selección del indicador Acoustic Diversity: La aplicación debe permitir seleccionar el indicador "Acoustic Diversity" para su procesamiento.	El sistema refleja que este indicador ha sido seleccionado y es incluido en el análisis.

Código	Descripción	Criterios de Calidad
RF005	Selección del indicador Acoustic Evenness: El investigador debe poder seleccionar el indicador "Acoustic Evenness".	El sistema muestra que este indicador ha sido marcado como seleccionado y aparece en los resultados.
RF006	Selección del indicador Bioacoustics: La aplicación debe incluir y permitir la selección del indicador "Bioacoustics" para análisis.	El investigador puede confirmar que ha sido seleccionado y que los datos relacionados son procesados.
RF007	Selección del indicador Normalized Difference Sound: Debe habilitarse la selección de este indicador como parte del análisis.	El sistema indica que este indicador ha sido agregado a los parámetros de análisis.
RF008	Selección del indicador Spectral Entropy: El sistema debe incluir este indicador y permitir su selección.	El investigador puede activarlo desde la interfaz y ver su impacto en los resultados.
RF009	Selección del indicador Temporal Entropy: La funcionalidad debe permitir activar este indicador como parte del conjunto a analizar.	El sistema lo muestra como activo y lo incluye en el procesamiento.
RF010	Selección del indicador Number of Peaks: Se debe permitir la inclusión de este indicador para el análisis.	El investigador lo selecciona desde la interfaz y lo ve reflejado en los resultados.
RF011	Selección del indicador Wave Signal to Noise Ratio: Esta funcionalidad habilita la inclusión de este indicador específico.	El sistema confirma su selección y muestra resultados relacionados.
RF012	Selección de carpeta a analizar: La aplicación debe permitir al investigador seleccionar una carpeta con archivos de audio para su análisis.	Al elegir la carpeta, el sistema carga los archivos y comienza el procesamiento.
RF013	Exportación de resultados a Excel: El sistema debe exportar los resultados del análisis en formato Excel.	El investigador puede abrir el archivo generado y verificar la información contenida.
RF014	Documentación de uso de la aplicación: Se debe incluir documentación accesible sobre cómo usar todas las funciones de la aplicación.	El investigador accede a un documento PDF, sitio web o ayuda integrada que explica las funcionalidades.
RF015	Recuperación ante corte de luz: El sistema debe guardar estados intermedios para reanudar desde el último punto seguro.	Tras un reinicio, el sistema detecta el estado anterior y ofrece continuar desde ahí.
RF016	Instalación sencilla del software: La aplicación debe contar con un instalador o instrucciones claras para su instalación.	El investigador puede instalar el software sin problemas técnicos siguiendo pasos básicos.

Código	Descripción	Criterios de Calidad
RF017	Cancelar y modificar índices: El usuario puede detener el proceso de análisis y modificar la selección de indicadores.	El investigador puede pausar, cambiar la configuración y continuar el análisis.
RF018	Conteo de audios procesados y finalización: El sistema debe mostrar cuántos audios han sido procesados y cuándo finaliza el análisis.	El investigador puede visualizar un contador en tiempo real y recibir una notificación al finalizar.

MVP 2

Código	Descripción	Criterios de Calidad
RF019	Descarga de archivos desde Arbimon: El sistema debe permitir al investigador autenticar su cuenta y descargar archivos de audio almacenados en Arbimon para su posterior análisis.	El usuario puede iniciar sesión, seleccionar los archivos deseados y confirmar que estos se han descargado correctamente al directorio local.
RF020	Conversión de FLAC a WAV: El sistema debe convertir archivos de audio del formato FLAC a WAV para facilitar su procesamiento.	Los archivos .flac seleccionados se convierten correctamente a .wav en la ubicación indicada, con validación de tamaño y duración para verificar integridad.
RF021	Elección del origen de análisis: El sistema debe permitir al usuario elegir entre descargar archivos desde Arbimon o trabajar con archivos previamente almacenados localmente.	El usuario puede seleccionar entre dos opciones de origen de datos y cargar los archivos correctamente desde la fuente elegida.
RF022	Reanudación de descarga ante fallo: En caso de interrupciones de red, el sistema debe permitir reanudar la descarga desde el último punto exitoso.	Se verifica que tras un fallo de conexión, el sistema no reinicia toda la descarga sino que continúa desde donde se interrumpió.
RF023	Procesamiento en hilos: El sistema debe ejecutar tareas como descarga, conversión o análisis de manera paralela mediante hilos para optimizar el tiempo de ejecución.	El usuario observa un uso eficiente de los recursos y múltiples tareas se ejecutan de forma simultánea sin bloquear el sistema.
RF024	Visualización de errores en ejecución: El sistema debe mostrar de manera clara en pantalla los errores o excepciones que ocurran durante la ejecución.	Errores se muestran en consola o en una interfaz amigable con mensaje descriptivo y código del error si aplica.

Código	Descripción	Criterios de Calidad
RF025	Selección de archivo CSV: El sistema debe permitir al usuario seleccionar un archivo CSV desde su equipo para ser procesado gráficamente.	El usuario puede navegar por el sistema de archivos, seleccionar un CSV válido y recibir confirmación de carga exitosa.
RF026	Análisis gráfico con gráficos de violín: El sistema debe generar gráficos de violín a partir de un CSV cargado previamente, visualizando la distribución de los datos.	El sistema muestra gráficos claros y legibles con ejes correctamente etiquetados y agrupación por categorías si aplica.
RF027	Manual de usuario: Se debe proporcionar un manual que explique cómo utilizar las nuevas funcionalidades del sistema paso a paso.	El documento está disponible en formato PDF o digital, es accesible, contiene ejemplos y responde a preguntas frecuentes.
RF028	Manual técnico para desarrolladores: Debe existir un documento con detalles técnicos del sistema: arquitectura, librerías utilizadas, estructura de carpetas y dependencias.	El manual incluye instrucciones para instalación, mantenimiento y desarrollo futuro, con código comentado y referencias técnicas.
RF029	Informe del trabajo realizado: Se debe generar un informe final que documente el proceso, decisiones tomadas, pruebas y resultados del desarrollo.	El informe presenta secciones claras, evidencia de avances, capturas de pantalla, resultados y conclusiones.
RF030	Diferenciación de índices por categorías: El sistema debe identificar y diferenciar los datos presentes en el CSV según sus categorías para su análisis.	El usuario puede visualizar que los datos han sido correctamente agrupados por categorías (e.g., especie, lugar, fecha) y reflejados en los gráficos.

MVP 3

Código	Descripción	Criterios de Calidad
US031	Mensajes de consola claros: El sistema debe mostrar mensajes de consola descriptivos, comprensibles y útiles durante la ejecución de los procesos.	El usuario puede interpretar fácilmente los mensajes de consola, comprendiendo el estado actual del sistema o los posibles errores.
US032	Gestión de sesiones: El sistema debe permitir cerrar sesión y autenticar nuevamente a un usuario con diferentes credenciales.	El usuario puede cerrar sesión e iniciar sesión con otra cuenta sin perder la estabilidad del sistema ni los datos cargados.

Código	Descripción	Criterios de Calidad
US033	Procesamiento de múltiples formatos de audio: El sistema debe ser capaz de aceptar y procesar archivos en formatos WAV, MP3 y FLAC.	El usuario puede cargar archivos en cualquiera de los formatos especificados y ver los resultados del análisis sin errores de compatibilidad.
US034	Análisis de audios de distintas longitudes: El sistema debe procesar correctamente archivos de audio sin importar su duración.	El usuario puede analizar archivos de audio cortos o largos obteniendo resultados válidos en ambos casos.
US035	Identificación de aves en audios: El sistema debe identificar y mostrar las especies de aves detectadas en los audios analizados.	El usuario puede visualizar un listado de las aves identificadas en cada audio procesado.
US036	Análisis de frecuencia por sitio: El sistema debe generar gráficos de calor que representen la actividad de las especies por sitio y frecuencia.	El usuario puede visualizar los gráficos de calor para interpretar patrones de actividad de aves en diferentes sitios.
US037	Detención del análisis en paralelo: El sistema debe permitir detener de forma segura y exitosa el análisis en paralelo de archivos en ejecución.	El usuario puede cancelar un análisis en curso y verificar que el sistema libera correctamente los recursos sin fallos.
US038	Uso eficiente de recursos: El sistema debe optimizar el uso de CPU, memoria y almacenamiento, evitando el consumo innecesario de recursos.	El usuario puede ejecutar el sistema en equipos de capacidades limitadas sin afectar significativamente el rendimiento general.
US039	Manual de usuario: Se debe entregar un manual que documente el uso de las nuevas funcionalidades de la aplicación.	El usuario puede consultar el manual para comprender el uso de cada funcionalidad sin necesidad de asistencia adicional.
US040	Manual técnico: Se debe entregar un manual técnico dirigido a desarrolladores, detallando la arquitectura, componentes y mantenimiento del sistema.	El equipo de desarrollo puede consultar el manual para realizar mantenimientos, mejoras o solucionar problemas de forma eficiente.
US042	Informe del trabajo realizado: Se debe generar un informe detallado de las actividades y desarrollos implementados durante el proyecto.	El usuario o administrador puede revisar el informe para conocer el alcance y cumplimiento de los objetivos del proyecto.

Capítulo 3

Arquitectura del sistema

3.1. Diseño general del sistema

Para el diseño de la aplicación se está utilizando una arquitectura MVC, dividiendo el sistema en tres partes fundamentales como lo son el Modelo, la Vista y el Controlador. En la sección de Vista, estarán todas las ventanas de la aplicación, en la sección de Modelo, estarán los objetos base de la aplicación y en la sección Controlador, se encontrarán objetos especializados en utilizar los objetos base y su interacción con la sección de vista.

3.1.1. Diagrama de clases

La siguiente imagen 3.1 es el diagrama de clases de la aplicación utilizando el estándar UML. En la sección de Vista tenemos las clases Main y Bienvenida, las cuales son las ventanas del sistema. En la sección de Controller se encuentran los objetos Analyzer, encargado de realizar el análisis de índices, y progress, encargado de saber cuales archivos llevo analizados. En la sección de modelo se encuentran AudioFile, que se encarga de modelar un archivo de audio, e Index, que guarda la información del indice analizado para un archivo de audio.

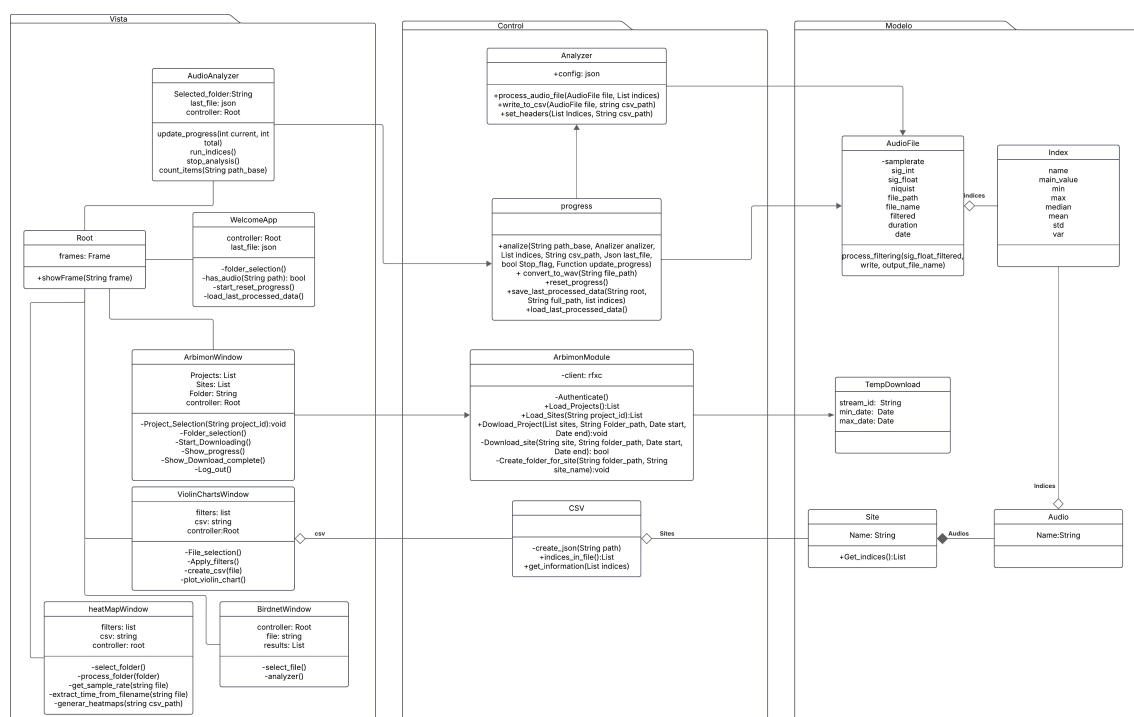


Figura 3.1: Diagrama de clases

Capítulo 4

Instrucciones de Instalación

4.1. Entorno de Desarrollo

4.1.1. Dependencias

El entorno de desarrollo utilizado está basado en Python, versión 3.0, específicamente la versión 3.10.11. Es necesario que el sistema tenga instalado el gestor de paquetes `pip`, el cual normalmente se incluye por defecto con las instalaciones modernas de Python. Las bibliotecas previamente instaladas para esta misma versión pueden llegar a dar problema, en caso de que esto suceda se recomienda borrar las bibliotecas instaladas y correr el comando de instalación proporcionado.

A continuación, se listan las dependencias necesarias y el orden en que deben instalarse:

- **Python3:** Debe de instalarse una versión de python 3.10, puede conseguirlo en el siguiente link: <https://www.python.org/downloads/release/python-31011/> o utilizando el script de instalación proporcionado.

4.1.2. Código fuente

Para descargar el código fuente se debe ir al siguiente enlace de github <https://github.com/JaimeCs27/PIPESOUND> le dará en la opción de code y luego seleccionará Download zip. Después de descargado el .zip debe descomprimirse en la ubicación que el usuario desee. Como se mencionó en la sección 4.1.1 debe de tener instalado Python 3. El código fuente trae un script de instalación que se encarga de instalar todas las dependencias necesarias, para correrlo puede ser de dos maneras. La primera es dándole doble click al archivo `installation.bat`.^o desde la línea de comandos corriendo el comando:

```
./Installation.bat
```

Este script se encargará de instalar python y pip en caso de que no esté instalado en el sistema, además instalará todas las bibliotecas necesarias para su correcto uso.

En caso de que el script de instalación presente algún problema, se detallará una guía para la instalación del sistema.

- **Paso 1:** Descargar el código fuente de la aplicación. Entrar la siguiente link de github <https://github.com/JaimeCs27/PIPESOUND>, luego presionar el botón que dice "cod", al presionarlo se abre un pequeño menú con una opción que dice "Download zip". Esto descargará el código de la aplicación.
- **Paso 2:** Descomprimir el archivo descargado en la ubicación que desee.

- **Paso 3:** Descargar python, ir al siguiente link: <https://www.python.org/downloads/release/python-31011/>, al abrir el link bajaremos a la sección de instalación, que luce como la imagen 4.1. En la cual elegiremos la versión "Windows instaler (64-bits)". Al presionar esto se descargará un ejecutable.

And now for something completely different

The Navier-Stokes equations are partial differential equations which describe the motion of viscous fluid substances, named after the French engineer and physicist Claude-Louis Navier and Anglo-Irish physicist and mathematician George Gabriel Stokes. They were developed over several decades of progressively building the theories, from 1822 (Navier) to 1842-1850 (Stokes).

The Navier-Stokes equations mathematically express momentum balance and conservation of mass for Newtonian fluids. They are sometimes accompanied by an equation of state relating to pressure, temperature and density. They arise from applying Isaac Newton's second law to fluid motion, together with the assumption that the stress in the fluid is the sum of a diffusing viscous term (proportional to the gradient of velocity) and a pressure term—hence describing the viscous flow. The difference between them and the closely related Euler equations is that Navier-Stokes equations take viscosity into account while the Euler equations model only inviscid flow. As a result, the Navier-Stokes are a parabolic equation and therefore have better analytic properties, at the expense of having less mathematical structure (e.g. they are never completely integrable).

[Full Changelog](#)

Files

Version	Operating System	Description	MD5 Sum	File Size	GPG	Sigstore
Gzipped source tarball	Source release		7e25e2f158b1259e271a45a249cb24bb	24.9 MB	SIG	sigstore
XZ compressed source tarball	Source release		1bf8481a683e9881e14d52e0f23633a6	18.7 MB	SIG	sigstore
macOS 64-bit universal2 installer	macOS	for macOS 10.9 and later	f5f7918e8bfb29f23860ab08712005	39.1 MB	SIG	sigstore
Windows installer (64-bit)	Windows	Recommended	a55e9c1e6421c84a4b8b4be41492f51	27.7 MB	SIG	sigstore
Windows installer (32-bit)	Windows		83a67e1c4f6f1472bf75d9681491bf1	26.6 MB	SIG	sigstore
Windows help file	Windows		52ff16ab5f300679889d3a93a8d50bb	9.0 MB	SIG	sigstore
Windows embeddable package (64-bit)	Windows		f1c0538b060e03cbb697ab3581cb73bc	8.2 MB	SIG	sigstore
Windows embeddable package (32-bit)	Windows		fee70dae9c25c60cbe825d6a1bfda57	7.3 MB	SIG	sigstore

[About](#)
[Downloads](#)
[Documentation](#)
[Community](#)
[Success Stories](#)
[News](#)

[Applications](#)
[All releases](#)
[Docs](#)
[Diversity](#)
[Arts](#)
[Python News](#)

[Quotes](#)
[Source code](#)
[Audio/Visual Talks](#)
[Mailing Lists](#)
[Business](#)
[PSF Newsletter](#)

Figura 4.1: Python version

- **Paso 4:** Instalar python. Ahora le daremos doble click al archivo descargado, que luce así 4.2. Le daremos un click a la opción en la zona inferior que dice "Add python.exe to PATH", Después seleccionaremos la opción de "Install Now" como se muestra en la figura 4.3

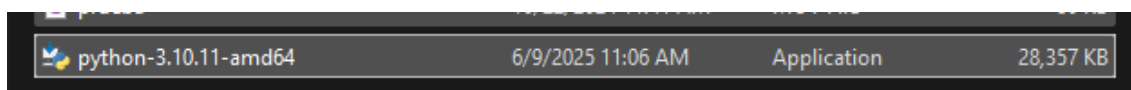


Figura 4.2: Python Installer



Figura 4.3: Python Installer

Después esperaremos hasta que el proceso de instalación se acabe. Al acabar saldrá una ventana como la siguiente 4.4 y nada más le daremos "close".

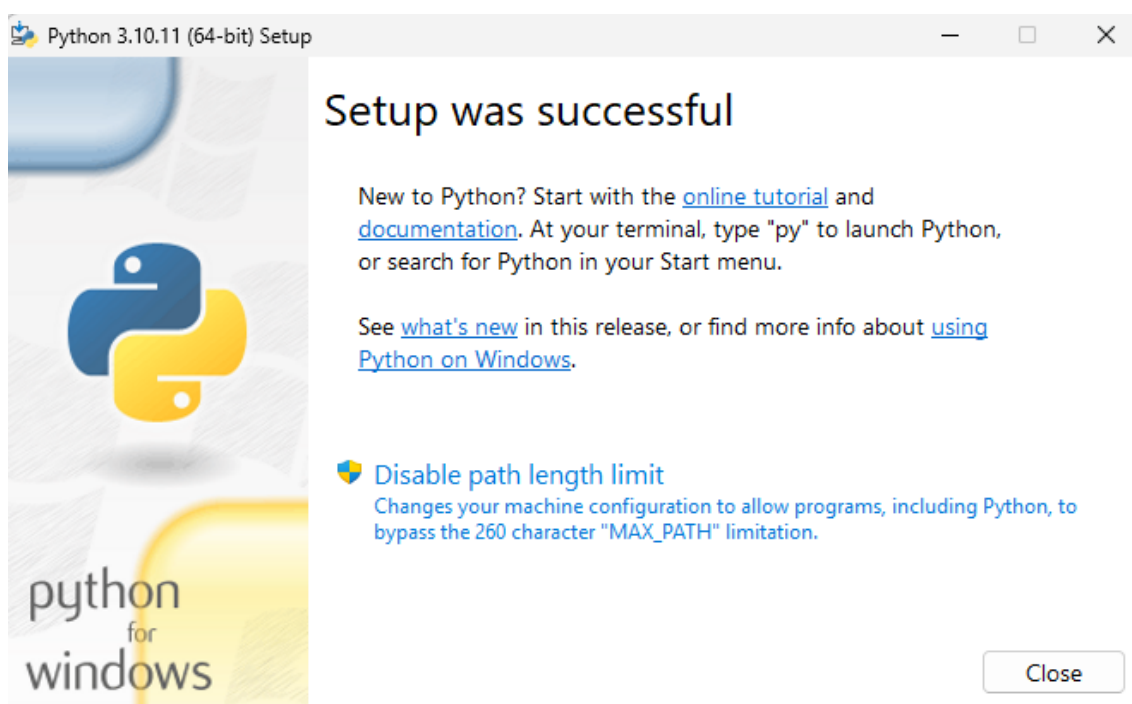


Figura 4.4: Python Installer

- **Paso 5:** Verificar la instalación, para verificarla abriremos una ventana de símbolo del sistema, presionando el boton de windows o en el buscador, y escribir "CMD", esto abrirá una ventana de comandos y escribiremos el siguiente comando:

```
python -V
```

Y saldrá lo siguiente 4.5

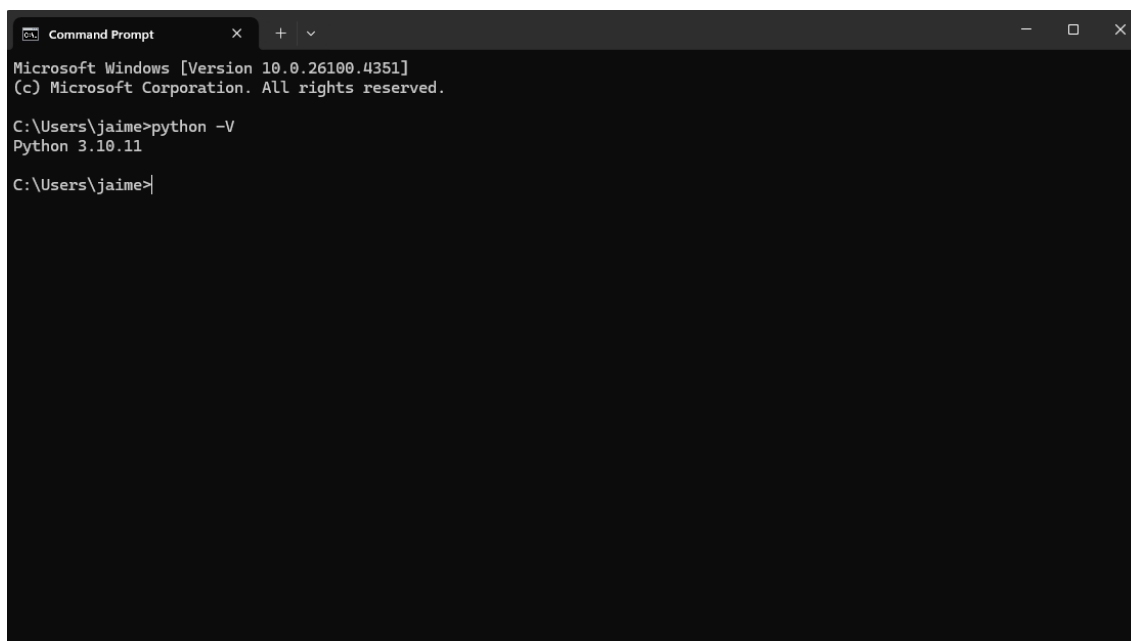


Figura 4.5: Verificación

- **Paso 6:** Instalar bibliotecas necesarias, para esto con la misma ventana de símbolo del sistema escribiremos el siguiente comando. Por motivos del documento se escribió dividido, pero debe ser una sola línea como se muestra en la figura 4.6

```
py -3.10 -m pip install customtkinter pillow scipy numpy pyyaml  
soundfile matplotlib seaborn librosa psutil requests_toolbelt  
httplib2 birdnetlib birdnet
```

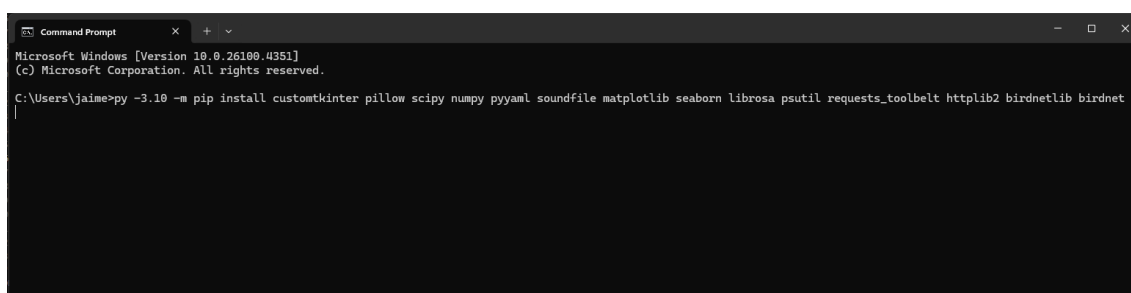


Figura 4.6: Instalación Bibliotecas

4.1.3. Despliegue en entorno de pruebas

Para el despliegue en un entorno de pruebas se debe tener descargado el código fuente mencionado anteriormente y correr el archivo de installation.bat, después de tener todas estas dependencias se puede correr el programa dando doble click sobre el archivo run.bat o por medio de consola escribir el siguiente comando:

```
./Run.bat
```

En caso de tener problemas con alguna de las bibliotecas a la hora de correr este comando, se recomienda leer el mensaje proporcionado por python y en caso de que falte alguna biblioteca, agregarla al script de instalación proporcionado y volver a correr este script.

4.2. Entorno de Producción

4.2.1. Dependencias

Por la naturaleza de la aplicación, las dependencias en producción son iguales a las dependencias de desarrollo, se necesita una versión de Python 3.10 en la computadora donde se va a ejecutar el programa junto con el paquete `pip`.

4.2.2. Instrucciones de instalación

Para instalar la aplicación primero debemos descargar el código fuente ubicado en el siguiente github: <https://github.com/JaimeCs27/PIPESOUND>. El código fuente trae en el dos archivos importantes los cuales son `installation.bat` y `run.bat`. Antes de intentar correr la aplicación debemos descargar e instalar las bibliotecas necesarias para esto tenemos el archivo `installation.bat` el cual al ejecutarse se encarga de descargar todas estas dependencias. Este archivo puede ejecutarse dándole doble click o desde la línea de comandos:

```
./Installation.bat
```

Después de que se terminen de instalar todas las bibliotecas, ya podremos utilizar el archivo `run.bat` para correr nuestro programa y probarlo.

En caso de tener problemas con el script de instalación referirse a la sección 4.1.4 de este documento para instalar el sistema.

4.3. Requerimientos del sistema

La aplicación PIPESOUND al realizar análisis de grandes cantidades de archivos, es altamente demandante en cuanto a recursos computacionales. En esta sección se especificarán requerimientos recomendados y mínimos para utilizar el programa.

Los requerimientos que se eligieron se basaron en las capacidades computacionales de los equipos en que se probaron todas las funcionalidades del sistema.

4.3.1. Requerimientos mínimos

- **CPU:** Procesador de 64-bits, 4 núcleos y 8 hilos, ya sea de intel o su contraparte AMD
- **Memoria:** 16 GB de ram
- **OS:** Windows 11

4.3.2. Requerimientos recomendados

- **CPU:** Procesador de 64-bits, 8 núcleos y 16 hilos, ya sea de intel o su contraparte AMD
- **Memoria:** 32 GB de ram
- **OS:** Windows 11

Capítulo 5

Conclusiones y trabajo futuro

5.1. Conclusiones

- El uso de la biblioteca de Python CustomTkinter resultó bastante importante para poder construir una interfaz moderna y funcional de manera sencilla. Esta biblioteca permitió crear componentes personalizables, los cuales mejoran la experiencia del usuario al utilizar la aplicación.
- Identificar los puntos seguros y los parámetros necesarios para guardar el estado de un programa en ejecución es fundamental para garantizar la continuidad y consistencia del flujo del mismo, porque estos puntos permiten recuperar el funcionamiento correcto ante errores o interrupciones inesperadas
- Gracias al diseño del prototipo visual de la aplicación, tanto el cliente como el equipo de desarrollo lograron obtener una visión más clara del objetivo final del proyecto, facilitando así la toma de decisiones, la alineación de expectativas y la planificación de los siguientes pasos a seguir
- El uso de hilos a la hora de ejecutar el análisis hizo que el rendimiento de la aplicación mejorara notablemente. Es un recurso muy útil para cuando se busque implementaciones eficientes. Utilizando archivos temporales para evitar regiones críticas en el flujo de trabajo.

5.2. Problemáticas y limitaciones

Una problemática del sistema es una diferencia pequeña de decimales que existe entre los resultados de la biblioteca de python acoustic index y la biblioteca de R soundecology.

El sistema operativo para el cual se desarrolló es Windows.

5.3. Trabajo futuro

Se puede ampliar el módulo de birdnet para poder entrenarlo con información propia y así ampliar el rango de especies o enfocarla en una región específica. Además puede generarse un CSV con los resultados del análisis de birdnet para poder utilizar la información en otro sistema.

Referencias bibliográficas

de Investigación e Innovación Tecnológica (LIIT), L. (s.f.). *Universidad estatal a distancia (uned) - sitio web institucional*. <https://investiga.uned.ac.cr/liit/>. (Consultado el 1 de mayo de 2025)