



**INFORME DE AVANCE
DEL PROYECTO**

**PIPELINE PARA INDICADORES DE SONIDOS EN
PYTHON**

**Para el curso de
PROYECTO DE INGENIERÍA DE SOFTWARE**

EQUIPO DE TRABAJO

JAIME CABEZAS SEGURA	COORDINADOR
NATASHA CALDERÓN ROJAS	DESARROLLADOR
SEBASTIÁN LÓPEZ VILLAVICENCIO	DESARROLLADOR
JOSÉ ANDRÉS RAMÍREZ CASTILLO	DESARROLLADOR
ROBERTO VARGAS MASIS	SUPERVISOR

**BAJO LA SUPERVISIÓN DE
RODOLFO MORA ZAMORA**

**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.

Índice general

1. Anteproyecto	2
1.1. Descripción y Alcance del proyecto	2
1.1.1. Antecedentes	2
1.1.2. Objetivos	2
1.1.3. Interesados	3
1.1.4. Estructura de Desglose de Trabajo	3
2. Especificación de requerimientos	4
2.1. Pila de producto	4
2.1.1. Requerimientos funcionales del sistema	4
2.1.2. Bitácora de cambios	6
2.2. Producto Mínimo Viable de la iteración 1	7
2.2.1. Pila de trabajo de la iteración 1	7
2.2.2. Desempeño del equipo	7
2.3. Producto Mínimo Viable de la iteración 2	8
2.3.1. Pila de trabajo de la iteración 2	8
2.3.2. Desempeño del equipo	9
2.4. Producto Mínimo Viable de la iteración 3	10
2.4.1. Pila de trabajo de la iteración 3	10
2.4.2. Desempeño del equipo	10
3. Arquitectura del sistema	12
3.1. Diseño general del sistema	12
3.1.1. Diagrama de clases	12
4. Conclusiones y trabajo futuro	13
4.1. Conclusiones	13
4.2. Problemáticas y limitaciones	13
4.3. Trabajo futuro	13
Referencias bibliográficas	14

Índice de figuras

2.1. Burndown chart	8
2.2. Burndown chart	9
2.3. Burndown chart	11
3.1. Diagrama de clases	12

Capítulo 1

Anteproyecto

1.1. Descripción y Alcance del proyecto

1.1.1. Antecedentes

El siguiente proyecto se realizará en colaboración con la Universidad Estatal a Distancia (UNED) y Laboratorio Investigación e Innovación Tecnológica (LIIT), los cuáles en conjunto han desarrollado proyectos enfocados en el paisaje acústico en distintas zonas del territorio nacional. El proyecto se enfoca en monitorear a través de placas ambientales llamados “AudioMoth”, alguna zona específica, generando un análisis que les permite determinar o desarrollar un paisaje sonoro del ecosistema.

Uno de los proyectos recientes es el de distribución espacio-temporal y ecología acústica del caracara avispero en la zona norte de Costa Rica. Cuyo objetivo era determinar la distribución espacio-temporal de una especie de ave para contribuir al entendimiento de su comportamiento, hábitat, interacciones y conservación de la especie.

En el proyecto actual están colaborando tanto estudiantes y profesores de la UNED, donde cada uno de los participantes se encarga de tomar su muestra de audio y subirlo a la plataforma de Arbimon. Estos audios luego son analizados y procesados respectivamente.

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.

1.1.2. 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.

Nombre del involucrado	Rol	Contacto
Roberto Vargas Masis 1	Cliente	Contacto
Adriana Céspedes Vindas	Contra parte técnica	Contacto

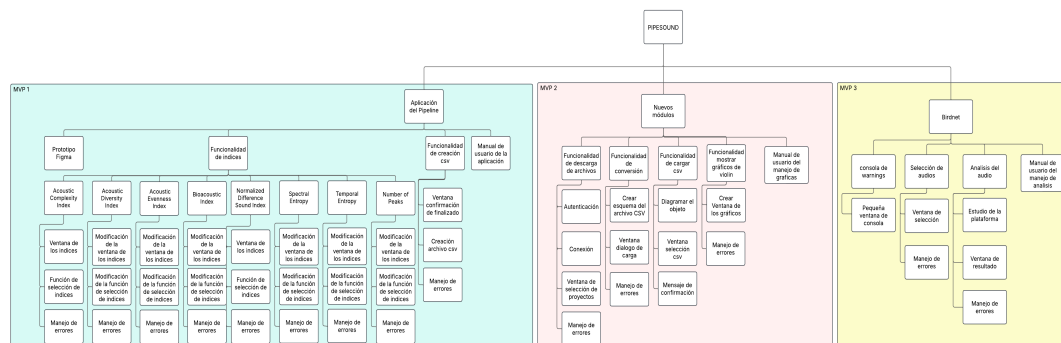
Cuadro 1.1: Involucrados

■ 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.

1.1.3. Interesados

1.1.4. Estructura de Desglose de Trabajo



Link al diagrama: Diagrama

Capítulo 2

Especificación de requerimientos

2.1. Pila de producto

Describa brevemente el producto en función del análisis de historias. Si es posible modifique la codificación de las historias para que el lector tenga un apoyo visual respecto a la estructura del sistema. Por ejemplo las historias correspondientes a los módulos de interfaz podrían codificarse UIXXX, mientras que las historias correspondientes a módulos de red podrían codificarse NTXXX. Explique en esta sección la lógica detrás de la codificación usada.

2.1.1. Requerimientos funcionales del sistema

Pila general del producto

Código	Descripción	Prioridad	Inserción
RF001	Yo como investigador quiero ver un prototipo de la aplicación	Alta	Original
RF002	Yo como investigador quiero ver todos los indicadores para seleccionarlos	Alta	Original
RF003	Yo como investigador quiero seleccionar el indicador de Acoustic Complexity para analizarlo	Alta	Original
RF004	Yo como investigador quiero seleccionar el indicador de Acoustic Diversity para analizarlo	Alta	Original
RF005	Yo como investigador quiero seleccionar el indicador de Acoustic Evenness para analizarlo	Alta	Original
RF006	Yo como investigador quiero seleccionar el indicador de Bioacustics para analizarlo	Alta	Original
RF007	Yo como investigador quiero seleccionar el indicador de Normalized difference Sound para analizarlo	Alta	Original
RF008	Yo como investigador quiero seleccionar el indicador de Spectral Entropy para analizarlo	Alta	Original
RF009	Yo como investigador quiero seleccionar el indicador de Temporal Entropy para analizarlo	Alta	Original

Código	Descripción	Prioridad	Inserción
RF010	Yo como investigador quiero seleccionar el indicador de Number of peaks para analizarlo	Alta	Original
RF011	Yo como investigador quiero seleccionar el indicador de Wave Signal to Noise Ratio para analizarlo	Alta	Original
RF012	Yo como investigador quiero poder elegir una carpeta para ser analizada	Media	Original
RF013	Yo como investigador quiero ver el análisis de los índices en un excel	Media	Original
RF014	Yo como investigador quiero tener documentación de cómo utilizar la aplicación	Baja	Original
RF015	Yo como investigador quiero que en caso de corte de luz, vuelva a analizar desde el último punto seguro	Media	Original
RF016	Yo como investigador quiero instalar el software fácilmente	Media	Original
RF017	Yo como investigador quiero poder cancelar el proceso para modificar los índices	Media	Original
RF018	Yo como investigador quiero saber cuantos audios llevo procesados y cuando termino el proceso	Baja	Original
RF019	Yo como investigador quiero ver el análisis de todos los índices en un excel	Alta	Original
RF020	Yo como investigador quiero descargar mis archivos desde arbimon para procesarlos	Alta	Original
RF021	Yo como investigador quiero convertir los archivos FLAC a WAV para procesarlos	Alta	Original
RF022	Yo como investigador quiero elegir entre descargar mis archivos por medio de arbimon o analizar archivos ya existentes en mi espacio de trabajo	Alta	Original
RF023	Yo como investigador quiero que en caso de fallas en el internet inicie la descarga desde el último punto seguro	Alta	Original
RF024	Yo como investigador quiero que el programa corra en hilos	Alta	Sprint 2
RF025	Yo como investigador quiero ver en pantalla los errores que suceden durante la ejecución	Media	Sprint 2
RF026	Yo como investigador quiero seleccionar un csv para analizarlo graficamente	Alta	Sprint 2
RF027	Yo como investigador quiero analizar por medio de graficos de violin un csv seleccionado	Alta	Sprint 2
RF028	Yo como investigador quiero tener un manual de usuario para las nuevas funcionalidades	Media	Sprint 2

Código	Descripción	Prioridad	Inserción
RF029	Yo como investigador quiero tener un manual técnico para los desarrolladores	Media	Sprint 2
RF030	Yo como investigador quiero tener un informe del trabajo realizado	Media	Sprint 2
RF031	Yo como investigador quiero diferenciar los índices por categorías	Baja	Sprint 2
RF031	Yo como investigador quiero diferenciar los índices por categorías	Baja	Sprint 2
US032	Yo como investigador quiero tener un claro entendimiento de los mensajes de consola	Media	Sprint 3
US033	Yo como investigador quiero poder cerrar sesión de arbimon y iniciar sesión con otra cuenta	Media	Sprint 3
US034	Yo como investigador quiero poder procesar archivos de distintos formatos como WAV, MP3, FLAC	Media	Sprint 3
US035	Yo como investigador quiero poder analizar audios de distintas longitudes	Alta	Sprint 3
US036	Yo como investigador quiero saber que aves se identifican en un audio determinado	Media	Sprint 3
US037	Yo como investigador quiero tener un análisis por sitio de frecuencia con graficos de calor	Media	Sprint 3
US038	Yo como investigador quiero detener con éxito el análisis en paralelo de los archivos	Media	Sprint 3
US039	Yo como investigador quiero que la aplicación no consuma más recursos de los necesarios	Alta	Sprint 3
US040	Yo como investigador quiero tener un manual de usuario para las nuevas funcionalidades	Alta	Sprint 3
US041	Yo como investigador quiero tener un manual técnico para los desarrolladores	Media	Sprint 3
US042	Yo como investigador quiero tener un informe del trabajo realizado	Media	Sprint 3

Cuadro 2.1: VERDE: Historias agregadas en esta iteración. ROJO: Historias eliminadas.

2.1.2. Bitácora de cambios

Se eliminó la historia RF019 debido a que en un principio se planeaba dividir en dos sprints el análisis de índices, esta historia era la encargada de mostrar los resultados de los índices del segundo sprint. Debido a que todos los índices se realizaron en el primer sprint, esta historia se volvió innecesaria.

2.2. Producto Mínimo Viable de la iteración 1

Desarrollar una aplicación que permita seleccionar entre los indicadores Acoustic Complexity Index, Acoustic Diversity Index, Acoustic Evenness Index, Bioacustics Index, Normalized Difference Sound Index, Spectral Entropy, Temporal Entropy, Number of Peaks y Wave Signal to Noise Ratio de manera interactiva por medio de una interfaz gráfica y generar un análisis de estos índices a un grupo seleccionado de grabaciones de audio, con la posibilidad de cancelar el proceso en cualquier momento en dado caso que el usuario se haya equivocado, y también ver el progreso del procesamiento de una manera visual en la aplicación, dando como resultado un archivo csv con dichos índices. Además crear un prototipo de figma que muestre el flujo del sistema y un manual de usuario con las indicaciones para utilizar completamente la aplicación.

2.2.1. Pila de trabajo de la iteración 1

Miembro del equipo	Rol	Capacidad
Jaime Cabezas Segura	Desarrollo	12
Natasha Calderón Rojas	Desarrollo	12
Sebastián López Villavicencio	Desarrollo	12
Jose Ramírez Castillo	Desarrollo	12
Presupuesto total:		48

Cuadro 2.2: Resumen de capacidad de carga del equipo para la iteración 1

Pila de la **iteración 1**

Código	CE	CR	Responsables	Finalización
RF001	1	1	Natasha Calderón	19/03/2025
RF002	3	2	Sebastian Lopez	26/03/2025
RF003	3	3	Jaime Cabezas	24/03/2025
RF004	3	3	Jose Ramírez	28/03/2025
RF005	3	3	Natasha Calderon	31/03/2025
RF006	3	2	Jaime Cabezas	28/03/2025
RF007	3	2	Sebastian Lopez	01/04/2025
RF008	3	2	Jaime Cabezas	01/04/2025
RF009	3	2	Jose Ramírez	03/04/2025
RF010	3	2	Natasha Calderon	02/04/2025
RF011	3	2	Sebas Lopez	07/04/2025
RF012	2	2	Jose Ramírez	09/04/2025
RF013	2	2	Jaime Cabezas	04/04/2025
RF014	2	2	Sebas Lopez	10/04/2025
RF015	3	3	Natasha Calderon	10/04/2025
RF016	2	2	Jaime Cabezas	08/04/2025
RF017	1	1	Jaime Cabezas	09/04/2025
RF018	1	1	Jaime Cabezas	09/04/2025
Total	44	36		

Cuadro 2.3: Pila de la Iteración 1. **CE:** Carga Estimada, **CR:** Carga Real.

2.2.2. Desempeño del equipo

El equipo no encontró dificultades durante el desarrollo de este primer sprint.

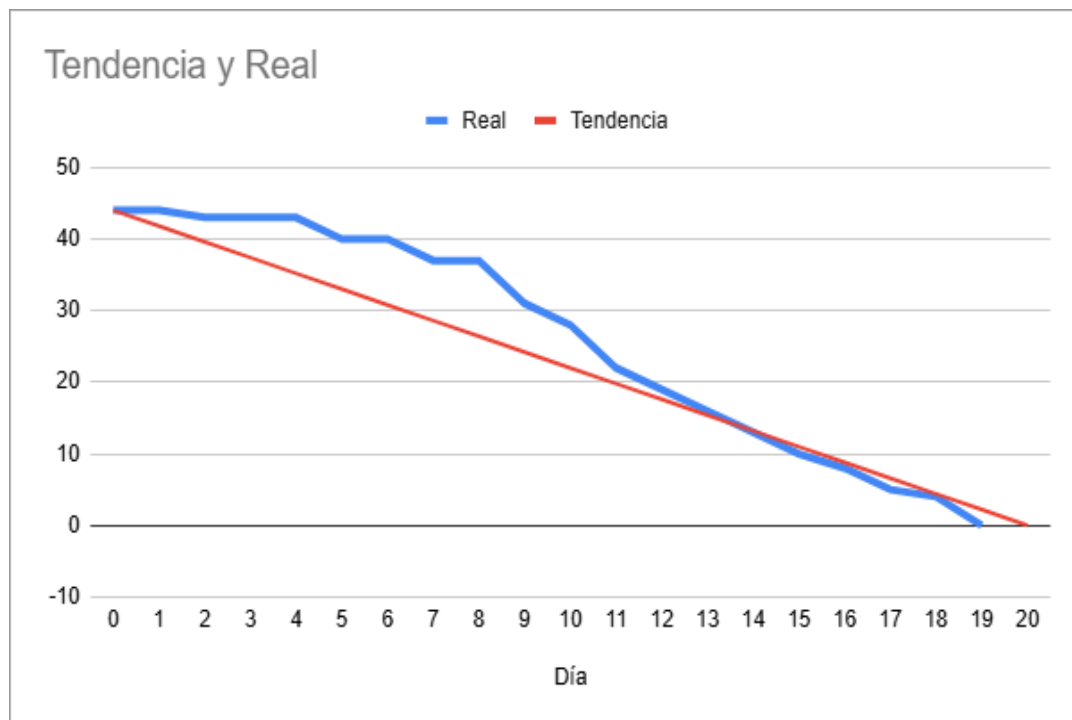


Figura 2.1: Burndown chart

Como podemos ver en el cuadro 2.7 la carga estimada era de 44 puntos de sprint y la carga real fue de 36 puntos de sprint, esto debido a que se sobreestimó la parte de integración de la biblioteca de acoustic-indices, la cual al principio parecía más complicado de lo que realmente fue, haciendo así que gran parte de los requerimientos que tenían relación con estos, terminaran con menos puntos de sprint de los planteados.

2.3. Producto Mínimo Viable de la iteración 2

Desarrollar las funcionalidades de descarga automática de las grabaciones de audio desde el sitio Arbimon para esto se requiere realizar la autenticación del usuario en arbimon, crear la conexión que permita mostrar la ventana de selección de proyectos y la conversión de los archivos a formato WAV. Además de la implementación de la funcionalidad que permite al usuario ver los datos analizados en formato de gráficos de violin, según el sitio y el índice seleccionado. Se debe desarrollar el manual de usuario sobre el funcionamiento de estas nuevas funcionalidades agregadas.

2.3.1. Pila de trabajo de la iteración 2

Miembro del equipo	Rol	Capacidad
Jaime Cabezas Segura	Desarrollo	12
Natasha Calderón Rojas	Desarrollo	12
Sebastián López Villavicencio	Desarrollo	12
Jose Ramírez Castillo	Desarrollo	12
Presupuesto total:		48

Cuadro 2.4: Resumen de capacidad de carga del equipo para la iteración 2

Pila de la iteración 2

Código	CE	CR	Responsables	Finalización
RF019	1			
RF020	6	6	Jaime Cabezas	05/05/2025
RF021	6	6	Natasha Calderon	05/05/2025
RF022	1	1	Natasha Calderon	04/24/2025
RF023	7	9	Jose Ramirez	05/12/2025
RF024	4	5	Sebastian Lopez	05/05/2025
RF025	4	3	Sebastian Lopez	05/09/2025
RF026	3	3	Jaime Cabezas	05/09/2025
RF027	4	3	Natasha Calderon	05/12/2025
RF028	2	2	Sebastian Lopez	05/15/2025
RF029	2	2	Jaime Cabezas	05/14/2025
RF030	2	2	Jose Ramirez	05/15/2025
RF031	2	1	Natasha Calderon	05/14/2025
Total	43	41		

Cuadro 2.5: Pila de la Iteración 2. CE: Carga Estimada, CR: Carga Real.

2.3.2. Desempeño del equipo

Tendencia y Real

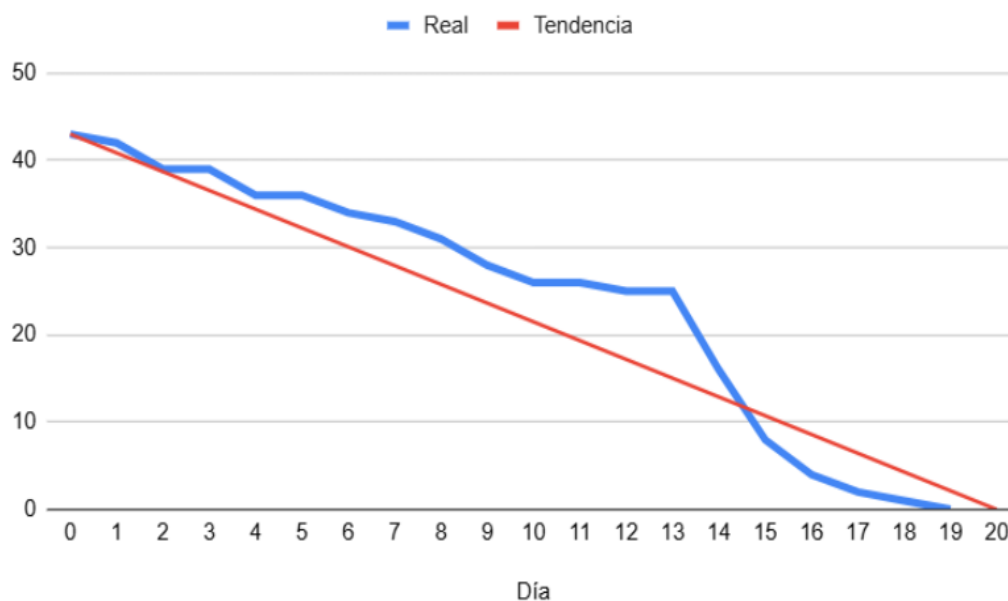


Figura 2.2: Burndown chart

En esta segunda iteración del sprint la carga estimada era de 43 puntos y la carga real fue de 41 puntos. En esta ocasión la estimación estuvo más acorde con la realidad del trabajo realizado y la mayoría de tareas desarrolladas tomaron la cantidad de puntos estimados.

2.4. Producto Mínimo Viable de la iteración 3

Desarrollar la funcionalidad que permite al usuario seleccionar una grabación o un grupo de grabaciones para determinar cuales especies de ave se encuentran registradas en el audio. También la funcionalidad de poder observar por medio de mapas de calor, un análisis visual de un sitio y como se comporta a lo largo de las horas. Además de un manual de usuario explicando la funcionalidad..

2.4.1. Pila de trabajo de la iteración 3

Miembro del equipo	Rol	Capacidad
Jaime Cabezas Segura	Desarrollo	12
Natasha Calderón Rojas	Desarrollo	12
Sebastián López Villavicencio	Desarrollo	12
Jose Ramírez Castillo	Desarrollo	12
Presupuesto total:		48

Cuadro 2.6: Resumen de capacidad de carga del equipo para la iteración 3

Pila de la iteración 3

Código	CE	CR	Responsables	Finalización
RF032	2	2	Sebastian Lopez	05/29/2025
RF033	2	2	Jaime Cabezas	05/23/2025
RF034	3	3	Natasha Calderon	05/27/2025
RF035	4	1	Jose Ramirez	06/02/2025
RF036	8	8	Jaime Cabezas	06/10/2025
RF037	4	3	Natasha Calderon	06/11/2025
RF038	3	3	Sebastian Lopez	06/10/2025
RF039	3	2	Jose Ramirez	06/12/2025
RF040	2	2	Sebastian Lopez	06/13/2025
RF041	2	2	Jaime Cabezas	06/13/2025
RF042	2	2	Jaime Cabezas	06/13/2025
Total	35	30		

Cuadro 2.7: Pila de la Iteración 3. **CE:** Carga Estimada, **CR:** Carga Real.

2.4.2. Desempeño del equipo

En esta última iteración del sprint la carga estimada era de 35 puntos y la carga real fue de 30 puntos. El equipo tuvo algunos problemas en la finalización del sprint debido a proyectos externos que tuvieron una influencia en el desarrollo de este proyecto.

Tendencia y Real

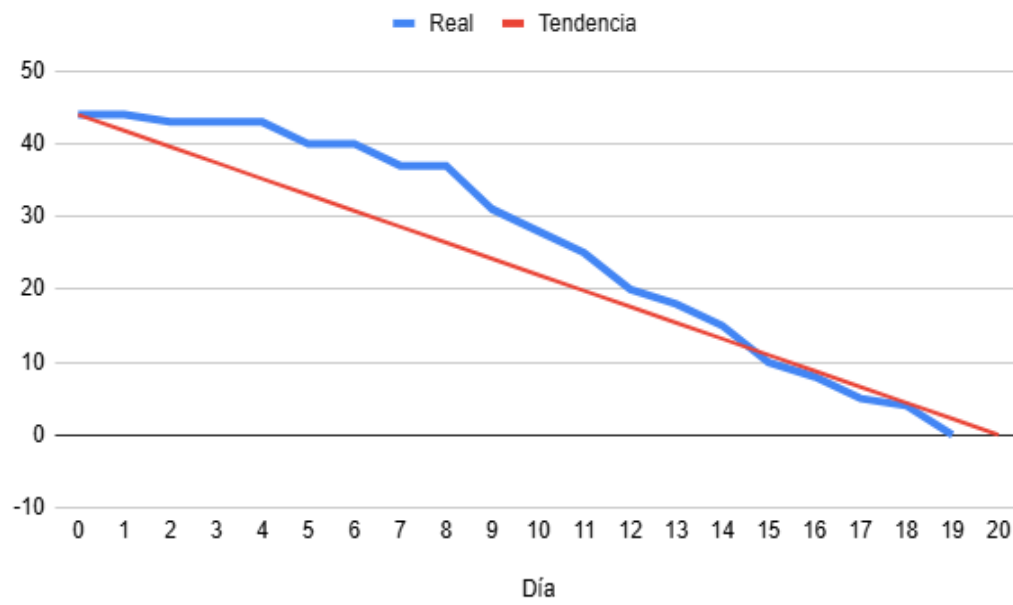


Figura 2.3: Burndown chart

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 Analizer, 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.

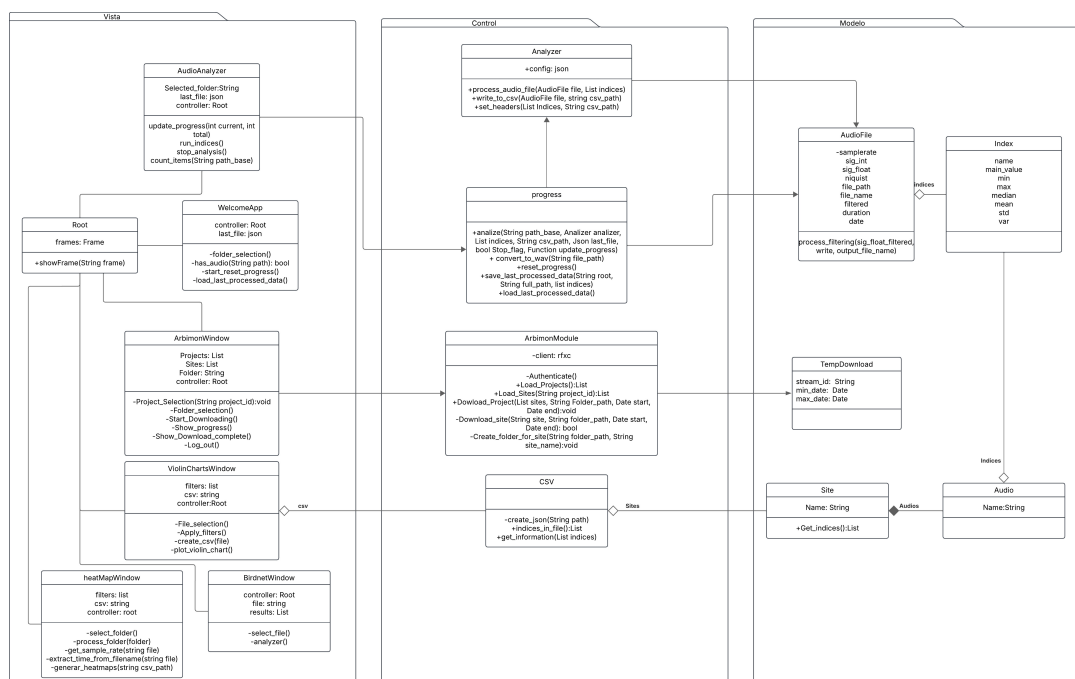


Figura 3.1: Diagrama de clases

Capítulo 4

Conclusiones y trabajo futuro

4.1. 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.
- Se utilizaron archivos temporales para evitar regiones críticas en el flujo de trabajo.

4.2. Problemáticas y limitaciones

- Una problemática del sistema es una pequeña diferencia 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.

4.3. Trabajo futuro

En un futuro, el proyecto podría incluir una mayor cantidad de índices para el análisis de audio y, además, categorizar diferentes especies de animales. Implementar un módulo robusto de BirdNet, donde se pueda entrenar una red neural y también cambiar entre las redes que proporciona la biblioteca.

Referencias bibliográficas