

UNIVERSIDAD TÉCNICA DEL NORTE



Facultad de Ingeniería en Ciencias Aplicadas Carrera de Software

Desarrollo de una aplicación de escritorio que automatice el respaldo de grabaciones de la programación radial utilizando C# y búsqueda de información en el audio mediante técnicas de inteligencia artificial.

Trabajo de grado previo a la obtención del título de Ingeniero de Software presentado ante la ilustre Universidad Técnica del Norte.

Autor:

Chocho Guandinango Daniel Mesias.

Director:

Ing. Rea Peñafiel Xavier Mauricio.

Ibarra – Ecuador

2024



UNIVERSIDAD TÉCNICA DEL NORTE

BIBLIOTECA UNIVERSITARIA

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE

LA UNIVERSIDAD TÉCNICA DEL NORTE

1. IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

| DATOS DE CONTACTO | | | |
|-----------------------------|---|------------------------|------------|
| CÉDULA DE IDENTIDAD: | 1004948871 | | |
| APELLIDOS Y NOMBRES: | CHOCHO GUANDINANGO DANIEL MESIAS. | | |
| DIRECCIÓN: | OTAVALO, SAN JUAN DE ILUMAN, SAN LUIS DE AGUALONGO. | | |
| EMAIL: | dmchochog@utn.edu.ec | | |
| TELÉFONO FIJO: | S/N | TELÉFONO MÓVIL: | 0969760323 |

| DATOS DE LA OBRA | |
|--------------------------------|--|
| TÍTULO: | DESARROLLO DE UNA APLICACIÓN DE ESCRITORIO QUE AUTOMATICE EL RESPALDO DE GRABACIONES DE LA PROGRAMACIÓN RADIAL UTILIZANDO C# Y BÚSQUEDA DE INFORMACIÓN EN EL AUDIO MEDIANTE TÉCNICAS DE INTELIGENCIA ARTIFICIAL. |
| AUTOR(ES): | CHOCHO GUANDINANGO DANIEL MESIAS |
| FECHA: | 18/11/2024 |
| PROGRAMA: | PREGRADO |
| TÍTULO POR EL QUE OPTA: | INGENIERO DE SOFTWARE |
| DIRECTOR: | MSc. REA PEÑAFIEL XAVIER MAURICIO. |
| ASESOR 1: | MSc. DAISY IMBAQUINGO |

I. CONSTANCIAS

El autor (es) manifiesta (n) que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es (son) el (los) titular (es) de los derechos patrimoniales, por lo que asume (n) la responsabilidad sobre el contenido de esta y saldrá (n) en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 18 días del mes de noviembre de 2024

EL AUTOR:

A handwritten signature in blue ink, appearing to read 'Daniel M. Chocho', is written over a horizontal dashed line.

ESTUDIANTE

Daniel Mesias Chocho Guandinango

C.I 1004617864

CERTIFICACIÓN DIRECTOR

Ibarra 15 de noviembre del 2024

CERTIFICACIÓN DIRECTOR DEL TRABAJO DE TITULACIÓN

Por medio del presente yo REA PEÑAFIEL XAVIER MAURICIO, certifico que el Sr. DANIEL MESIAS CHOCHO GUANDINANGO portador de la cedula de ciudadanía número 1004948871, ha trabajado en el desarrollo del proyecto de grado **“Desarrollo de una aplicación de escritorio que automatice el respaldo de grabaciones de la programación radial utilizando C# y búsqueda de información en el audio mediante técnicas de inteligencia artificial.”**, previo a la obtención del Título de Ingeniero en Software realizado con interés profesional y responsabilidad que certifico con honor de verdad.

Es todo en cuanto puedo certificar a la verdad

Atentamente



Firmado electrónicamente por:
XAVIER MAURICIO REA
PEÑAFIEL

Ing. Rea Peñafiel Xavier Mauricio
DIRECTOR DE TRABAJO DE GRADO

DEDICATORIA

Dedico este trabajo y el esfuerzo invertido en su realización a mi familia, quienes siempre estuvieron presentes brindándome su apoyo incondicional, en especial a mis padres (María Clara Guandinango y José Rafael Chocho). Gracias a su incansable esfuerzo y aliento, pude mantenerme enfocado y resiliente durante todo este proceso.

Asimismo, extendiendo esta dedicación a mis compañeros que, con determinación, se mantuvieron constantes hasta alcanzar nuestros objetivos comunes, superando juntos las dificultades.

Daniel Mesias
Chocho Guandinango

AGRADECIMIENTO

Dedico este trabajo de tesis a mi familia, quienes han sido mi pilar fundamental durante todo este proceso académico, brindándome su apoyo incondicional y motivación constante.

A mis queridos padres, María Clara Guandinango y José Rafael Chocho, por su amor inquebrantable, sacrificio y enseñanzas. Su sabiduría y fortaleza me han guiado y fortalecido a lo largo de mi vida.

Al director de tesis, Ing. Mauricio Rea, y a mi asesora, MSc. Daisy Imbaquingo, por su invaluable guía y conocimientos. Su dedicación y paciencia han sido cruciales para la realización de este trabajo. Agradezco sus orientaciones, que han sido fundamentales para la calidad y éxito de esta investigación.

A la empresa RS 105.5 FM, por abrirme sus puertas y proporcionar un entorno propicio para el desarrollo de mi investigación. Su colaboración y disposición para facilitarme la información necesaria han sido vitales para la culminación de esta tesis. Gracias por permitirme aplicar mis conocimientos en un entorno real y por su apoyo durante este proceso.

A la Carrera de Ingeniería en Software, por ofrecerme las herramientas y conocimientos necesarios para crecer profesionalmente. La formación recibida ha sido integral y me ha preparado para enfrentar los retos del mundo laboral. Agradezco a todos los docentes y administrativos por su compromiso con nuestra educación y desarrollo.

Finalmente, a mis compañeros, por su camaradería, apoyo y colaboración a lo largo de estos años. Juntos hemos superado muchas dificultades y compartido innumerables momentos de aprendizaje y crecimiento. Su amistad y compañerismo han hecho de este camino una experiencia enriquecedora y memorable.

Daniel Mesias
Chocho Guandinango

Tabla de Contenido

| | |
|--|------|
| DEDICATORIA..... | v |
| AGRADECIMIENTO | vi |
| ÍNDICE DE FIGURAS..... | xi |
| ÍNDICE DE TABLAS | xii |
| RESUMEN | xiii |
| ABSTRACT..... | xiv |
| INTRODUCCIÓN | 15 |
| Tema..... | 15 |
| Problema | 15 |
| Planteamiento de problema..... | 15 |
| Objetivos..... | 17 |
| Objetivo general..... | 17 |
| Objetivos específicos. | 17 |
| Alcance. | 17 |
| Metodología | 18 |
| CAPÍTULO 1..... | 20 |
| 1.1. La radio y la tecnología..... | 20 |
| 1.2. Desarrollo de la aplicación para automatizar las grabaciones de audio..... | 22 |
| 1.2.1. Metodologías..... | 22 |
| 1.2.2. Metodología TDD:..... | 23 |
| 1.3. Lenguaje de programación C#..... | 30 |

| | | |
|-----------------|---|----|
| 1.3.1. | Grabación de audio en C#..... | 30 |
| 1.3.2. | Librerías para la grabación de audio en C#. | 31 |
| 1.4. | Inteligencia artificial en el procesamiento de audio | 32 |
| 1.4.1. | Técnicas de búsqueda de información en audio con IA. | 33 |
| 1.4.2. | Speech Recognition. | 34 |
| 1.4.3. | Características de las técnicas de Speech Recognition. | 34 |
| 1.4.4. | ¿Cómo funciona el Speech Recognition? | 34 |
| 1.5. | Norma ISO/IEC 25010 | 35 |
| 1.5.1. | Definición de usabilidad | 36 |
| 1.5.2. | Características de usabilidad..... | 36 |
| 1.5.3. | Principios de diseño centrados en el usuario | 36 |
| 1.5.4. | Métodos de evaluación de usabilidad | 36 |
| 3.1. | Trabajos relacionados | 36 |
| CAPÍTULO 2..... | | 40 |
| 2.1. | Diseño de lógica del negocio. | 40 |
| 2.1.1. | Gestión de grabaciones. | 40 |
| 2.1.2. | Automatización de las grabaciones de respaldo. | 40 |
| 2.1.3. | Búsqueda y análisis de audio. | 41 |
| 2.1.4. | Arquitectura de la aplicación de escritorio | 41 |
| 2.2. | Configuración del entorno de desarrollo..... | 42 |
| 2.2.1. | Instalación y configuración..... | 42 |

| | | |
|-----------------|---|----|
| 2.2.1.5. | Creación del proyecto. | 45 |
| 2.3. | Maquetado de la interfaz de usuario. | 46 |
| 2.3.1. | Prototipado de la interfaz de usuario en Windows Forms. | 46 |
| 2.4. | Desarrollo de la aplicación de escritorio aplicando TDD y POO | 47 |
| 2.4.1. | Método listar dispositivos test. | 47 |
| 2.4.2. | Método listar dispositivos implementación. | 48 |
| 2.4.3. | Método activar evento test. | 49 |
| 2.4.4. | Método activar evento implementación. | 50 |
| 2.4.5. | Aplicación de POO en C#..... | 51 |
| 2.4.6. | Integración de Speech Recognition | 52 |
| 2.4.7. | Script Básico de Python con modelo Speech Recognition | 52 |
| 2.4.8. | Integración del modelo Speech Recongition. | 53 |
| CAPÍTULO 3..... | | 55 |
| 3.2. | Metodología de evaluación. | 55 |
| 3.2.1. | Criterios de usabilidad. | 55 |
| 3.2.2. | Instrumentos de evaluación..... | 56 |
| 3.2.3. | Población y muestra..... | 56 |
| 3.3. | Procedimiento de Evaluación. | 58 |
| 3.3.1. | Plan de pruebas. | 58 |
| 3.3.2. | Recolección de datos..... | 58 |
| 3.3.3. | Análisis de datos. | 58 |

| | |
|---|----|
| 3.3.4. Resultados de encuesta. | 59 |
| 3.3.5. Análisis y discusión. | 64 |
| 3.4. Conclusiones..... | 66 |
| 3.5. Recomendaciones | 66 |
| Bibliografía..... | 68 |
| Anexos | 73 |
| Anexo 1. Entrevista de la situación actual..... | 73 |
| Anexo 2. Cuestionario de Usabilidad basado en la ISO 25010..... | 76 |
| Anexo 3: Software de grabación automática en ejecución..... | 81 |
| Anexo 4: Certificado de entrega del software final..... | 82 |

ÍNDICE DE FIGURAS

| | |
|---|----|
| Figura 1. Árbol de problemas. | 16 |
| Figura 2. Arquitectura física del funcionamiento de la aplicación. | 18 |
| Figura 3. Ciclo de desarrollo TDD. | 26 |
| Figura 4. Arquitectura del modelo Speech Recognition. | 35 |
| Figura 5. Arquitectura de la aplicación de escritorio. | 42 |
| Figura 6. Instalación de Visual Studio Community 2022. | 43 |
| Figura 7. Instalación de .NET | 44 |
| Figura 8. Instalación de paquete NAudio | 44 |
| Figura 9. Instalación de NUnit..... | 45 |
| Figura 10. Creación del proyecto y la solución C# .NET..... | 45 |
| Figura 11. Instalación de Microsoft Windows Desktop Runtime 6.0.9..... | 46 |
| Figura 12. Prototipo de la interfaz de usuario en Windows Forms..... | 47 |
| Figura 13 Método listar dispositivos de entrada disponibles Test. | 48 |
| Figura 14. Método listar dispositivos de entrada disponibles Implementación..... | 49 |
| Figura 15. Método activar evento Test. | 50 |
| Figura 16. Método activar evento implementación. | 51 |
| Figura 17. Script Básico de Python con modelo Speech Recognition..... | 53 |
| Figura 18. Integración del modelo Speech Recongnition..... | 54 |

ÍNDICE DE TABLAS

| | |
|---|----|
| Tabla 1. Situación actual de los recursos para la grabación. | 21 |
| Tabla 2. Situación actual de la actividad humana para la grabación. | 21 |
| Tabla 3. Parámetros de automatización. | 40 |
| Tabla 4. Parámetros de búsqueda y análisis de audio. | 41 |
| Tabla 5. Entradas, actividades y salidas de la aplicación. | 41 |
| Tabla 6. Clases y métodos principales. | 51 |
| Tabla 7. Subcaracterísticas e indicadores de usabilidad. | 55 |
| Tabla 8. Distribución de la población objetivo. | 56 |
| Tabla 9. Justificación de la Distribución de la Muestra. | 57 |
| Tabla 10. Comprensión de la interfaz del Software de grabación. | 59 |
| Tabla 11. Uso de las funciones principales del Software de grabación. | 59 |
| Tabla 12. Navegación dentro Software de grabación. | 60 |
| Tabla 13. Eficiencia de la funcionalidad de realizar respaldos de audio. | 60 |
| Tabla 14. Búsqueda de información en el audio de respaldo. | 61 |
| Tabla 15. Tiempo para completar tareas. | 61 |
| Tabla 16. Errores o dificultades técnicas mientras usaba el Software de grabación. ... | 62 |
| Tabla 17. Estética y diseño visual del Software de grabación. | 62 |
| Tabla 18. Resiliencia a fallos. | 62 |
| Tabla 19. Recomendación a otros usuarios. | 63 |
| Tabla 20. Criterio de evaluación para la característica de usabilidad. | 64 |
| Tabla 21. Análisis y aceptación de Usabilidad. | 65 |

RESUMEN

La presente tesis, titulada "Desarrollo de una aplicación de escritorio que automatice el respaldo de grabaciones de la programación radial utilizando C# y técnicas de inteligencia artificial", tiene como objetivo crear una herramienta que facilite la gestión y respaldo de grabaciones radiales mediante tecnologías avanzadas de procesamiento de audio. Primero, se desarrollará un marco teórico que abarcará conceptos clave como la captura de audio en C# y la transcripción de audio a texto usando el modelo pre-entrenado Speech Recognition. También se profundizará en el uso de la metodología de Desarrollo Guiado por Pruebas (TDD) con NUnit, proporcionando una base sólida para la implementación del sistema. El segundo objetivo es el desarrollo de una aplicación en C# que automatice el proceso de respaldo de grabaciones radiales, integrando el modelo de transcripción y la metodología TDD. Esto permitirá gestionar de manera eficiente los archivos de audio. Finalmente, se evaluará la usabilidad de la aplicación, uno de los atributos clave de la norma ISO/IEC 25010 sobre la calidad del software. Esta evaluación identificará fortalezas y oportunidades de mejora, asegurando que la herramienta cumpla con los estándares de calidad esperados y brinde una óptima experiencia de usuario. En conclusión, esta tesis busca no solo ofrecer una solución técnica para el respaldo automatizado de grabaciones radiales, sino también contribuir al desarrollo de software mediante la aplicación de metodologías y estándares internacionales de calidad.

ABSTRACT

This thesis, titled "Development of a Desktop Application for Automating the Backup of Radio Program Recordings Using C# and Artificial Intelligence Techniques," aims to create a tool that facilitates the management and backup of radio recordings through advanced audio processing technologies. First, a theoretical framework will be developed, covering key concepts such as audio capture in C# and audio-to-text transcription using the pre-trained Speech Recognition model. Additionally, the implementation of Test-Driven Development (TDD) with the NUnit framework will be explored, providing a solid foundation for the system's development. The second objective is to develop a desktop application in C# that automates the backup process of radio recordings, integrating the transcription model and TDD methodology. This will allow for efficient management of audio files. Finally, the usability of the developed application will be evaluated, focusing on this key attribute according to the ISO/IEC 25010 software quality standard. The evaluation will identify strengths and improvement opportunities, ensuring that the tool meets expected quality standards and delivers an optimal user experience. In conclusion, this thesis not only aims to provide a technical solution for the automated backup of radio recordings but also contributes to the field of software development by applying internationally recognized methodologies and quality standards.

INTRODUCCIÓN

Tema

Desarrollo de una aplicación de escritorio que automatice el respaldo de grabaciones de la programación radial utilizando C# y búsqueda de información en el audio mediante técnicas de inteligencia artificial.

Problema

Planteamiento de problema

La relación entre radio y tecnología potencia la comunicación al aplicar capacidades digitales a procesos. Mejora la eficiencia, transparencia, inclusividad, valor al cliente y servicios (Mosquera Aguilar, 2022). La radio, como medio de comunicación, está regulada por la "Ley Orgánica de Comunicación" (Lema & Romero, 2023). La ley impone obligaciones, algunas son problemas para la radio RS 105.5 FM, como la clasificación de audiencias y franjas horarias (Art. 65), así como la obligación de archivo de soportes (Art. 91) para los medios audiovisuales (Art. 74) (LEY ORGÁNICA DE COMUNICACIÓN, 2019).

Respetar franjas horarias es crucial; infringir ofende al oyente, exponiendo a contenidos inapropiados y explícitos (CORDICOM, 2014). La "Ley Orgánica de Comunicación" exige cumplimiento, su violación lleva a sanciones legales y económicas (Perugachi, 2023).

Los programas en vivo y entrevistas sociales pueden causar controversias. Contar con un respaldo permite abordar conflictos, ofrecer réplicas o buscar soluciones entre las partes afectadas, si es necesario (Morales, 2023).

La dimensión comercial se suma al problema: las empresas publicitarias requieren informes precisos de menciones en la radio. La falta de respaldos de audio afecta el cobro, la credibilidad y la confianza con los clientes (Perugachi, 2023).

La radio RS 105.5 FM enfrenta estos problemas debido a la carencia de respaldos de grabaciones de audio y de un software adecuado, las grabaciones se hacen de manera manual con un programa ineficiente desperdiciando los recursos de la PC, siendo ineficiente en la realización de informes y defensas legales (Morales, 2023). Árbol de problemas en la figura 1.

Figura 1.

Árbol de problemas.



Objetivos

Objetivo general.

Desarrollar de una aplicación de escritorio de gestión automática de respaldo de grabaciones de la programación radial con búsqueda de información en el audio utilizando C# y un modelo pre- entrenado de inteligencia artificial.

Objetivos específicos.

- Elaborar un marco teórico sobre la captura de audio mediante C# y la transcripción de audio a texto con Speech Recognition, y la implementación de la metodología TDD con NUnit.
- Desarrollar una aplicación de escritorio aplicando la metodología TDD con NUnit, en el lenguaje C# y el modelo pre- entrenado Speech Recognition.
- Evaluar el rendimiento de la aplicación, centrándose en el atributo de usabilidad de la calidad de Software establecido en la ISO/IEC 25010.

Alcance.

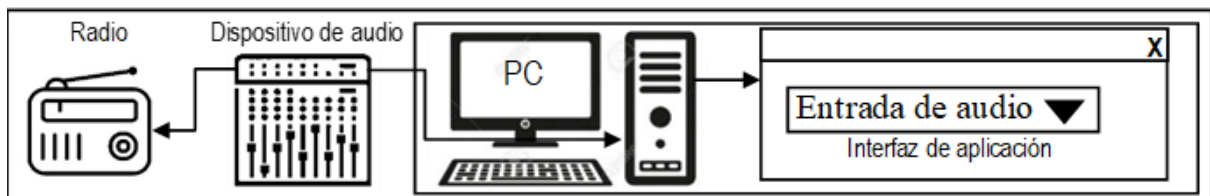
El presente trabajo de investigación se basa en desarrollar una aplicación de escritorio para gestionar respaldo de grabaciones de audio en la radio RS 105.5 FM, abordando los siguientes puntos:

- a. La aplicación se desarrollará en C#, se ejecutará en Windows 10 y 11 (64 bits) sobre el entorno de ejecución .NET 6.0 o posterior, figura 2.
- b. Reconocimiento de información clave dentro del audio posterior a la grabación en un segmento de máximo 2 horas y mapeo referencial utilizando el modelo pre- entrenado (Speech Recognition) de IA.
- c. El desarrollo de la aplicación será utilizando las pruebas unitarias (TDD) con NUnit.

- d. Grabaciones desde entrada física o lógica la misma que sale al aire en la radio, formato .mp3, canal estéreo, tasa de bits seleccionable (92-320 kbps), figura 3.
- e. Parámetros; canal de audio, hora inicio, hora fin, días específicos, prioridad (inmediata o retrasada) y lista de palabras claves para el mapeo de información dentro del audio.
- f. Características adicionales: arranque con Windows, guardar, actualizar, borrar evento, grabación manual y mensajes de alerta (advertencias, confirmaciones, etc.).
- g. La aplicación prescinde de bases de datos y, utiliza la ruta del sistema para almacenar las grabaciones.
- h. La aplicación entrará en producción durante una semana. La validación del aplicativo se llevará a cabo en base al atributo de usabilidad de la calidad de Software establecido en la norma ISO/IEC 25010.

Figura 2.

Arquitectura física del funcionamiento de la aplicación.



Metodología

Tipo de investigación: aplicada.

- 1 **Para el objetivo 1:** Se recopilará información de artículos científicos, trabajos de titulación, libros, etc., de la Universidad Técnica del Norte, entrevista al personal de la radio RS 105.5 FM y la documentación oficial de (NUnit, C#, Speech Recognition).

- 2 **Para el objetivo 2:** Se aplicará la metodología TDD, el paradigma de programación POO, el modelo pre-entrenado Speech Recognition y en el lenguaje C#, dentro del IDE Visual Studio.
- 3 **Para el objetivo 3:** La validación se llevará a cabo en base al atributo de usabilidad de la calidad de Software establecido en la norma ISO/IEC 25010.

CAPÍTULO 1.

1.1. La radio y la tecnología.

La rapidez con que la vida sufre transformaciones es progresivamente asombrosa. Las tecnologías emergentes transforman el funcionamiento de los medios de comunicación. Los medios sufren cambios, transformaciones y maduraciones. A este tema se han dedicado numerosas charlas y análisis escritos, ya que las (TIC) tecnologías de la información y la comunicación asumen un gran impacto en la sociedad de la información (Radios de América, 2022). La correlación entre los avances en comunicación y tecnología siempre ha sido evidente. Las TIC no sólo han impulsado la radio, sino que también la han revolucionado, convirtiéndola en una nueva plataforma que combina formatos antiguos con métodos modernos de distribución de contenidos. Las TIC han avanzado no sólo en la transmisión de contenidos vía radio, sino también en soluciones informáticas que ayudan a resolver problemas relacionados con los procesos radiofónicos, incluyendo la producción, la automatización y la planificación, entre otras áreas (Martín Pena & Parejo Cuellar, 2018).

La relación entre radio y tecnología potencia la comunicación al aplicar capacidades digitales a procesos. Mejora la eficiencia, transparencia, inclusividad, valor al cliente y servicios (Mosquera Aguilar, 2022). La radio, como medio de comunicación, está regulada por la "Ley Orgánica de Comunicación" (Lema & Romero, 2023). La ley impone obligaciones, algunas son problemas para la radio RS 105.5 FM, como la clasificación de audiencias y franjas horarias (Art. 65), así como la obligación de archivo de soportes (Art. 91) para los medios audiovisuales (Art. 74) (LEY ORGÁNICA DE COMUNICACIÓN, 2019). Respetar franjas horarias es crucial; infringir ofende al oyente, exponiendo a contenidos inapropiados y explícitos (CORDICOM, 2014). La "Ley Orgánica de Comunicación" exige cumplimiento, su violación lleva a sanciones legales y económicas (Perugachi, 2023).

Los programas en vivo y entrevistas sociales pueden causar controversias. Contar con un respaldo permite abordar conflictos, ofrecer réplicas o buscar soluciones entre las partes afectadas, si es necesario (Morales, 2023). La dimensión comercial se suma al problema: las empresas publicitarias requieren informes precisos de menciones en la radio. La falta de respaldos de audio afecta el cobro, la credibilidad y la confianza con los clientes (Perugachi, 2023).

La radio RS 105.5 FM enfrenta estos problemas debido a la carencia de respaldos de grabaciones de audio y de un software adecuado, las grabaciones se hacen de manera manual con un programa ineficiente desperdiciando los recursos de la PC, siendo ineficiente en la realización de informes y defensas legales (Morales, 2023). Árbol de problemas en la figura.

A continuación, se detalla la situación actual de la empresa radial en cuanto a la gestión de las grabaciones e insumos Tabla 1 y Tabla 2 (Morales, 2023).

Tabla 1.

Situación actual de los recursos para la grabación.

| Recurso | Productividad | Usabilidad | Medición en % | | |
|---------------------|---------------|------------|----------------|--------------|--------------|
| | | | Disponibilidad | Confiabledad | Satisfacción |
| Hardware | 80% | 80% | 80% | 80% | 80% |
| Software | 40% | 60% | 70% | 50% | 50% |
| Archivo de respaldo | 50% | 50% | 65% | 70% | 50% |

Tabla 2.

Situación actual de la actividad humana para la grabación.

| Actividad humana | Productividad | Usabilidad | Medición en % | | |
|-------------------------------|---------------|------------|----------------|--------------|--------------|
| | | | Disponibilidad | Confiabledad | Satisfacción |
| Iniciar grabación | 50% | 80% | 80% | 70% | 50% |
| Finalizar grabación | 50% | 80% | 80% | 70% | 50% |
| Automatización | Ninguna | Ninguna | Ninguna | Ninguna | 0% |
| Obtener información del audio | 50% | 70% | 60% | 60% | 40% |
| Realizar informes | 50% | 60% | 65% | 70% | 40% |

1.2. Desarrollo de la aplicación para automatizar las grabaciones de audio.

El desarrollo de software de escritorio implica la creación de programas destinados a ser ejecutados en una computadora de escritorio y desarrollados específicamente para abordar un requisito de usuario particular, en este caso, automatizar las grabaciones de los respaldos de la programación radial (Euroinnova, 2023). El software de escritorio sirve para un sin número de propósitos, como el procesamiento de textos, administración de hojas de cálculo, creación y edición de gráficos (De Cervantes Saavedra, 2021). El desarrollo de software para aplicaciones de escritorio a menudo incluye los lenguajes de programación como Java, C++, Python, C#, entre otros, para construir un programa autónomo que se puede instalar en una computadora (TECH Education, 2023).

El desarrollo de software de aplicaciones de escritorio es un procedimiento complejo que incluye el diseño y la creación de programas destinados a ejecutarse en una computadora de escritorio, ofreciendo a los clientes soluciones y capacidades personalizadas (Parra & Ramírez, 2018).

1.2.1. Metodologías.

La metodología de desarrollo se refiere al conjunto de técnicas empleados para crear una solución de software, gestionar el proceso de desarrollo y mitigar la aparición de errores. Es crucial para fines organizativos. Hay múltiples opciones disponibles, por lo que depende de cada equipo tomar una decisión sobre cuál elegir (Maida & Pacienza, 2015).

El empleo de una metodología permite economizar tiempo y mejorar la administración de los recursos disponibles, ya sea a corto o largo plazo, la optimización de los recursos dentro del presupuesto es uno de los pilares fundamentales de cualquier metodología (Flores, 2018).

Además, es esencial destacar el valor añadido de las metodologías más onerosas, estas pueden contribuir significativamente a la mejora del resultado final. El profesional a cargo del desarrollo de software desempeña diversas funciones, siempre siguiendo las directrices

proporcionadas por el cliente y comunicando avances o incidencias. En una primera instancia, se encarga de elaborar el código que sustentará la herramienta. Luego, procede a crear el software, abarcando el diseño de la interfaz. Por último, realiza pruebas exhaustivas para verificar la fiabilidad de la herramienta (Flores, 2018).

La elección de una determinada metodología dependerá de distintos factores como el orden, el tiempo, el presupuesto y la importancia de controlar riesgos. En este trabajo de investigación se utilizará el TDD.

1.2.2. Metodología TDD:

Las pruebas unitarias se realizan previo al desarrollo del código en una técnica de programación conocida como desarrollo guiado por pruebas (TDD). El objetivo principal es escribir código simple, limpio y resistente. Para hacer esto, primero se deben crear, probar y reescribir casos de prueba para cada funcionalidad que desee desarrollar si se encuentra una falla. El objetivo es que el desarrollo se acelere (a pesar de lo que inicialmente se pueda creer dado el tiempo asignado para ello). La duplicación de código es esencialmente el objetivo del enfoque TDD, que también permite a los desarrolladores escribir código nuevo sólo en caso de que las pruebas fallen. Esto corrige los numerosos errores que pueden surgir en miles de líneas de código y hace que la documentación sea mucho más transparente, lo que a su vez proporciona una red de seguridad ante cualquier modificación (Macas et al., 2021).

1.2.2.1. Desafíos.

La ingeniería de software es actualmente un campo destacado que gira en torno a la creación de aplicaciones por parte de expertos. Esta tecnología ha mejorado significativamente la calidad de vida humana. Por lo tanto, es crucial afrontar el desafío de desarrollar un producto de alta calidad que satisfaga los requisitos del cliente y promueva la difusión de estas tecnologías. Los modelos de las metodologías ágiles desempeñan un papel crucial a la hora de facilitar el cambio y mejorar el proceso de desarrollo de software. Priorizan las necesidades

cambiantes del cliente o usuario final, permitiendo realizar modificaciones a lo largo del ciclo de desarrollo. Este enfoque garantiza la creación de soluciones personalizadas. Además, existen modelos que pretenden potenciar este procedimiento desde su inicio, como el modelo que se basa en pruebas unitarias antes de empezar la producción del producto final. Es beneficioso contar con herramientas que automaticen el proceso de diseño, revisión y verificación de pruebas unitarias para fuentes de proyectos. Actualmente se han desarrollado varias herramientas para facilitar y optimizar el proceso de programación para diferentes lenguajes. Estas herramientas están ganando popularidad y se ha establecido fuertemente el mercado (Macas et al., 2021).

Esta investigación tiene como objetivo explorar más a fondo la metodología de desarrollo de software apoyado en pruebas (TDD) y resaltar el papel crucial del marco de programación extrema en su desarrollo. El marco de programación extremo facilita la gestión del desarrollo de software, mientras que TDD se centra específicamente en el diseño de sistemas. TDD, o Test Driven Development, surgió como un derivado de la metodología XP con el objetivo de optimizar el proceso de desarrollo, aumentar la productividad del equipo de desarrollo y lograr productos duraderos, mantenibles y de alta calidad. Se implementa antes de la finalización de las fuentes del proyecto y principalmente enfatiza la prueba de las unidades de software. El marco de desarrollo basado en pruebas ha ganado un reconocimiento significativo debido a su eficacia demostrada como herramienta de diseño en modelos corporativos. Además, se puede demostrar que este marco posee la capacidad de crear productos de mucha calidad (Corredor et al., 2017).

Una extensa investigación indica que el desarrollo basado en pruebas (TDD) ofrece ventajas significativas para las empresas de desarrollo de software. Las entrevistas realizadas como parte de la investigación revelan que la metodología TDD facilita la creación de código de alta calidad. Al escribir código inicialmente de manera simple y organizada, los

desarrolladores obtienen una comprensión integral de la lógica empresarial subyacente. TDD también permite la aplicación de modelos de calidad e inculca entre los desarrolladores la práctica de comprender a fondo el sistema antes de embarcarse en la codificación, evitando así la programación sin rumbo. Además, TDD mejora la productividad de los equipos de desarrollo al aumentar gradualmente su producción a medida que avanzan en el proceso algorítmico. En consecuencia, se mejora la comunicación efectiva entre las partes interesadas del proyecto (Macas et al., 2021).

La técnica de desarrollo de software basado en pruebas es un marco sólido cuya popularidad ha ido creciendo constantemente. Varios estudios han enfatizado la importancia de utilizar este enfoque para su adopción en proyectos realizados por empresas de desarrollo. Dentro del ámbito del desarrollo de software, existe un grupo de personas que no sólo son innovadoras, sino que también poseen un fuerte deseo de compartir información. Estas personas se esfuerzan por crear enfoques, métodos de enseñanza y tecnologías que profundicen en la teoría detrás del marco y dilucidan las ventajas de su implementación. La importancia de una metodología en las empresas radica en la optimización de los procesos y en conocimientos necesarios que debe poseer el equipo de desarrollo para utilizar eficazmente esta metodología (Corredor et al., 2017).

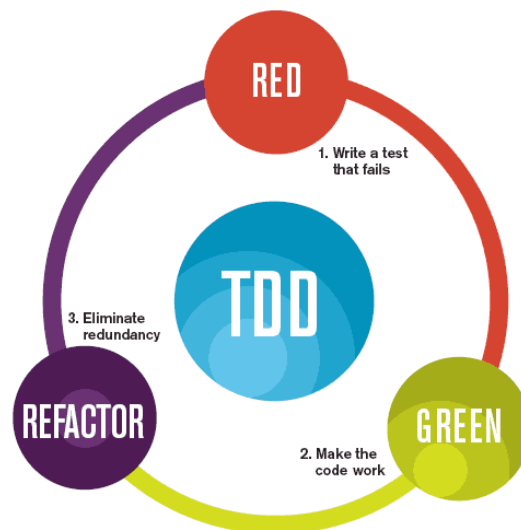
1.2.2.2. Ciclo de Desarrollo TDD.

- ✓ **Desarrollo de prueba:** Durante esta fase, el desarrollador debe establecer con precisión los criterios necesarios para completar el proceso de desarrollo del código. Estos requisitos servirán como pautas para codificar y validar diversas pruebas. Normalmente, la redacción de este proceso tendrá errores porque aún no se han implementado las medidas necesarias para su verificación. La fase roja es un nombre alternativo para este estado, ya que se caracteriza por la visualización de fallas reportadas en este tono, Figura 3 (Veintimilla & Cuenca, 2014).

- ✓ **Escritura del código:** La fase posterior consiste en implementar el código que se utilizará para validar las distintas pruebas (Veintimilla & Cuenca, 2014).
- ✓ **Validación de pruebas:** Una vez terminada las pruebas validadas, podemos confirmar el cumplimiento de los requisitos especificados. En esta fase no se tienen en cuenta las buenas prácticas. En cambio, la atención se centra únicamente en verificar que las pruebas prescritas se completen con éxito. Por eso se la conoce como fase verde, figura 1., ya que el verde es el tono más simbólico de este procedimiento (Veintimilla & Cuenca, 2014).
- ✓ **Refactorización:** En esta sección, el código escrito se someterá a una exploración para garantizar el desempeño de las mejores prácticas y validar su limpieza (Veintimilla & Cuenca, 2014).

Figura 3.

Ciclo de desarrollo TDD.



Nota: 1.1(SHIFT ASIA, 2024)

1.2.2.3. Beneficios de TDD:

- ✓ La mejora en la eficacia del producto de software se evidencia al eliminar redundancias y lograr una comprensión integral desde las etapas iniciales del proceso.

- ✓ Se logra la creación de código altamente reutilizable al desarrollar funcionalidades que pueden ser aprovechadas en diversas clases.
- ✓ La colaboración dentro del equipo de desarrollo se vuelve más cómoda, facilitando las relaciones interpersonales.
- ✓ Se fomenta la confianza entre las partes del equipo, incluso si algunos tienen menos experiencia, al proporcionar facilidades para ello.
- ✓ La comunicación con los miembros del equipo se ve multiplicada, fortaleciendo la interacción y coordinación.
- ✓ Los responsables de la garantía de calidad asumen un papel más inteligente e interesante en el proceso.
- ✓ La práctica de escribir el Test antes que el código nos induce a desarrollar la funcionalidad mínima necesaria, evitando el exceso de diseño.
- ✓ Al revisar un proyecto ejecutado mediante TDD, se evidencia que los testas actúan como una documentación práctica excelente, brindando una referencia clara para comprender el código.
- ✓ La productividad del equipo de desarrollo experimenta un aumento significativo.
- ✓ Se identifican y abordan más casos de uso durante la fase de diseño del software, contribuyendo a su robustez y adaptabilidad.

1.1.1.4. Pruebas unitarias.

Son secciones de código concisas y bien organizadas dentro de un módulo de software. Sirven para validar el funcionamiento esencial de las funciones del sistema. Las pruebas son un componente integral del diseño de software que define los requisitos del código y ayuda a validar sus resultados. El desarrollo de una prueba unitaria empieza con la definición de los requisitos del sistema (historias de usuario). El próximo paso consiste en codificar la prueba unitaria, donde se realizan validaciones y verificaciones para garantizar que la funcionalidad

escrita cumpla con las expectativas del cliente. El requisito se cumple según lo especificado y se resuelven los problemas identificados, como códigos redundantes, duplicaciones y fallas en la agrupación de funciones para pruebas unitarias (Laucaur & Eduardo, 2017).

Las pruebas unitarias se pueden automatizar, es decir, los lenguajes de programación cuentan con su propia tecnología para este fin. Hay soluciones disponibles que permiten la validación automatizada de pruebas unitarias directamente desde el código fuente, eliminando la necesidad de ejecución humana (panucar & Eduardo, 2017). Se pueden realizar varias pruebas unitarias para garantizar que la especificación narrativa del usuario sea completa. Las pruebas unitarias son beneficiosas de múltiples maneras. No sólo aceleran el proceso de desarrollo de software, sino que también facilitan su uso en otros proyectos, lo que permite un progreso rápido. Las pruebas unitarias se desarrollan de forma autónoma a partir de los componentes del proyecto, asegurando que no interrumpan la resolución final del sistema. Según algunos estudios, existen acusaciones de que el proceso de implementación de pruebas unitarias es lento. Sin embargo, cabe señalar que escribir un fragmento de prueba unitaria normalmente no lleva más de cinco 5 minutos (Fermin González et al., 2015).

1.1.1.5. Ventajas de aplicar pruebas unitarias.

- Las pruebas unitarias permiten el ordenamiento y la reestructuración del código y promueven la integración de nuevos módulos del sistema. Al permitir que se realicen modificaciones en unidades pequeñas sin afectar a todo el sistema, cualquier inconsistencia descubierta durante el desarrollo del sistema se puede corregir rápidamente (Jerez, 2017).
- Desarrollar pruebas unitarias bajo un estricto límite de tiempo de cinco (5) minutos y validar exhaustivamente cada prueba permite la detección oportuna de fallas en el código y la pronta resolución de cualquier problema que pueda surgir. Esto elimina la

necesidad de reiniciar el ciclo de desarrollo de software, lo que resulta en operaciones más rápidas (Jerez, 2017).

- Realizar pruebas unitarias desde el principio del proyecto y seguir el enfoque correcto permite identificar inconsistencias, refactorizar el código y eliminar la redundancia en el código fuente. Al finalizar el proyecto, obtendrá un resultado de alta calidad junto con un código meticulosamente estructurado y escrito de manera eficiente (Jerez, 2017).
- A medida que avanza el desarrollo de software, la implementación de pruebas unitarias permite proveer información valiosa para proyectos futuros. También sirve como medida preventiva, asegurando que los problemas encontrados en desarrollos anteriores no afecten a los nuevos. Además, las pruebas unitarias facilitan la rápida adopción de prácticas de desarrollo orientadas a pruebas, fomentando así una ciencia de diseño de software sólido (Jerez, 2017).
- Iniciar el ciclo de vida del desarrollo de software con pruebas unitarias, permite establecer un contexto de sistema integral, un énfasis preciso en el diseño y una determinación amplia de los requisitos necesarios para lograr un resultado de alta calidad (Jerez, 2017).

1.1.1.6. Herramientas para automatizar TDD.

Cada lenguaje de programación ofrece una variedad de herramientas y marcos diseñados específicamente para el proceso de desarrollo basado en pruebas (TDD). En esta instancia, el trabajo de estudio utilizará NUnit, un marco de prueba integrado con C# y .Net, como lenguaje de desarrollo. NUnit es un marco integral de pruebas unitarias diseñado para ser compatible con todos los lenguajes .Net. La versión de producción actual 3 del software ha sido objeto de una reescritura exhaustiva, incorporando numerosas capacidades adicionales

específicas del marco .NET, después de haber sido adaptada originalmente de Junit (*NUnit.Org*, 2023).

1.3. Lenguaje de programación C#.

C# es un lenguaje contemporáneo que se basa en objetos y garantiza la seguridad de tipos. Los desarrolladores pueden diseñar una extensa gama de aplicaciones seguras y resistentes que funcionan en la plataforma .NET. C# se deriva de la familia C de lenguajes de programación, lo que lo hace reconocible instantáneamente para los programadores familiarizados con C, C++, Java y JavaScript (Acosta, 2017).

C# está orientado a componentes y a objetos. Tiene características de lenguaje específicos que facilitan netamente estos conceptos, lo que transforma en un lenguaje intuitivo para crear y utilizar componentes de software. Desde sus inicios, ha incorporado funcionalidades para adaptarse a cargas de trabajo novedosas y metodologías de diseño de programas en evolución. C# es esencialmente un lenguaje orientado a objetos (Acosta, 2017).

1.3.1. Grabación de audio en C#.

C# es un lenguaje de programación muy ventajoso para grabar audio, y se convierte en una excelente opción para diseñar aplicaciones que impliquen grabación y procesamiento de sonido. C# es un lenguaje potente y orientado a objetos que facilita la creación de programas modulares y de fácil mantenimiento. La integración de este software con el ecosistema .NET ofrece a los usuarios acceso a una extensa gama de bibliotecas y herramientas, como NAudio, una biblioteca ampliamente utilizada para manipular datos de audio (Llamas, 2023).

La compatibilidad de C# para grabar audio es robusta y versátil. C# se ejecuta principalmente en el entorno de ejecución de .NET, que proporciona una capa de abstracción sobre el sistema operativo subyacente. Esto resulta en una alta portabilidad y compatibilidad multiplataforma, permitiendo que las aplicaciones desarrolladas en C# y .NET se ejecuten en diversos sistemas operativos, como Windows, Linux y macOS (Llamas, 2023).

La biblioteca NAudio, utilizada dentro del ecosistema C#, proporciona funcionalidad de captura de audio específicamente para computadoras con Windows. Aunque la compatibilidad de varios dispositivos de audio puede variar según las características y controladores del hardware, NAudio simplifica el proceso de interacción con interfaces de audio convencionales (Llamas, 2023). Además, el desarrollo mejorado de .NET Core y su progresión hacia .NET 5 y versiones posteriores ha reforzado significativamente la interoperabilidad multiplataforma. Con las últimas versiones de .NET Core/.NET 5+, tiene la capacidad de crear programas de grabación de audio en C# y ejecutarlos sin problemas en varios sistemas operativos, sin encontrar ningún problema de compatibilidad notable (Ibarra & Hurtado, 2022).

1.3.2. Librerías para la grabación de audio en C#.

C# tiene varias bibliotecas disponibles para la grabación de audio, para este propósito usaremos NAudio, un kit de herramientas de audio .NET de código abierto. NAudio ofrece una extensa gama de clases diseñadas específicamente para manipular archivos y dispositivos de audio en el sistema operativo Windows (Microsoft, 2023). Sus principales características son:

- ✓ **Grabación y reproducción de sonido:** NAudio simplifica la captura y reproducción de audio hacia y desde dispositivos de entrada/salida de audio, como micrófonos y altavoces.
- ✓ **Gestión de formatos de sonido:** facilita el trabajo con varios formatos de audio, incluidos WAV, MP3 y otros, y proporciona funciones para la conversión entre diferentes formatos de audio.
- ✓ **Efectos de sonido:** Admite la aplicación de efectos de sonido como ecualización, cambio de tono y reverberación, entre otros.

- ✓ **Representación visual del sonido:** Proporciona herramientas para visualizar datos de audio, como formas de onda y espectrogramas, lo que facilita el análisis y la representación gráfica.
- ✓ **Manipulación de secuencias de sonido:** Permite trabajar con secuencias de audio, lo que resulta útil para la transmisión en tiempo real y la manipulación eficaz de grandes conjuntos de datos de audio.
- ✓ **Secuenciación MIDI:** Incluye funcionalidades para trabajar con secuenciadores MIDI, lo que permite crear y reproducir música utilizando instrumentos virtuales.
- ✓ **Compatibilidad con WASAPI, ASIO y DirectSound:** proporciona compatibilidad con varias API de audio de Windows, como WASAPI, ASIO y DirectSound, y proporciona opciones avanzadas de configuración y rendimiento.
- ✓ **Compatibilidad con .NET Core y .NET 5+:** se ha actualizado con compatibilidad con .NET Core y versiones posteriores (.NET 5 y versiones posteriores), lo que hace posible el desarrollo de aplicaciones multiplataforma.
- ✓ **Participación de la comunidad:** Cuenta con una comunidad dinámica de desarrolladores que contribuyen al proyecto y brindan soporte, asegurando actualizaciones periódicas y mejoras continuas.
- ✓ **Licencia de código abierto:** NAudio se distribuye bajo la licencia MIT, lo que permite su uso en proyectos comerciales y de código abierto de forma gratuita.

1.4. Inteligencia artificial en el procesamiento de audio

La inteligencia artificial (IA) en el procesamiento de audio implica el uso de algoritmos y modelos sofisticados para comprender, analizar y manipular datos de audio automáticamente. La integración de la IA abarca múltiples dominios e influye profundamente en nuestras interacciones con el sonido en diversas facetas de la vida diaria y la tecnología (Iriz González, 2019).

El reconocimiento automático de voz (ASR) permite la conversión de señales de audio en texto, simplificando así la transcripción automática y la interacción mediante comandos de voz. Los modelos generativos permiten la creación de audio utilizando técnicas como la síntesis de voz y la generación de música. Además, los algoritmos de procesamiento del lenguaje natural (PLN) aumentan la comprensión del significado subyacente al habla hablada (Evin, 2019).

La IA se utiliza para mejorar la calidad del audio, analizar emociones, identificar sonidos ambientales y traducir audio automáticamente. Estas aplicaciones demuestran cómo la IA está revolucionando el uso del audio en diferentes industrias, incluidas la música, la seguridad y la atención médica (Iriz González, 2019).

1.4.1. Técnicas de búsqueda de información en audio con IA.

La recuperación de información de audio implica el uso de diversas metodologías para identificar fragmentos específicos de contenido dentro de las grabaciones de audio. Las técnicas abarcan una amplia gama de métodos, incluido el reconocimiento automatizado del habla (ASR), la indexación de características acústicas y la utilización de algoritmos de búsqueda de audio comparables (Pastor, 2022). También se tienen en cuenta los métodos que utilizan transcripciones automatizadas, modelos de incrustación de audio y análisis de sentimientos y emociones. Incorporar motores de búsqueda de texto, utilizar palabras clave y frases clave e implementar filtros especializados son tácticas complementarias que se realizan en este procedimiento (Carrasco, 2023).

La búsqueda de información en audio se basa en la integración de muchas tecnologías, que abarcan la conversión de audio en texto, así como el examen de atributos acústicos y emocionales. Esto facilita la utilización de herramientas adaptativas para discernir y recuperar información particular dentro de archivos de audio (Pastor, 2022).

1.4.2. **Speech Recognition.**

El objetivo primordial de este trabajo es utilizar la transcripción de audio a texto. Esta técnica es crucial en diversas situaciones, que van desde facilitar el acceso a personas con discapacidad auditiva hasta mejorar la productividad en entornos corporativos y académicos, específicamente para este trabajo en el contexto de los medios de radio (pypi.org, 2024)

El paquete Python **Speech Recognition** es una biblioteca de código abierto que facilita la creación de sistemas de reconocimiento de voz. Mantenido por la comunidad de código abierto, SpeechRecognition es versátil y admite múltiples motores de reconocimiento de voz, incluidos Google Web Speech API, Microsoft Bing Voice Recognition, IBM Speech to Text, entre otros (pypi.org, 2024).

1.4.3. **Características de las técnicas de Speech Recognition.**

- ✓ **Ponderación Lingüística:** Aumentar la precisión dando mayor importancia a palabras específicas que se mencionan frecuentemente (como nombres de productos o términos técnicos), además de los términos que ya están en el vocabulario básico (Boullosa, 2014).
- ✓ **Etiquetado de Oradores:** Crear una transcripción que identifique o etiquete las intervenciones de cada participante en una conversación con múltiples oradores (Boullosa, 2014).
- ✓ **Adaptación Acústica:** Entrenar el sistema para ajustarse a las características acústicas del entorno (como el ruido ambiental en un centro de llamadas) y a los estilos de habla (como el tono de voz, el volumen y el ritmo) (Boullosa, 2014).

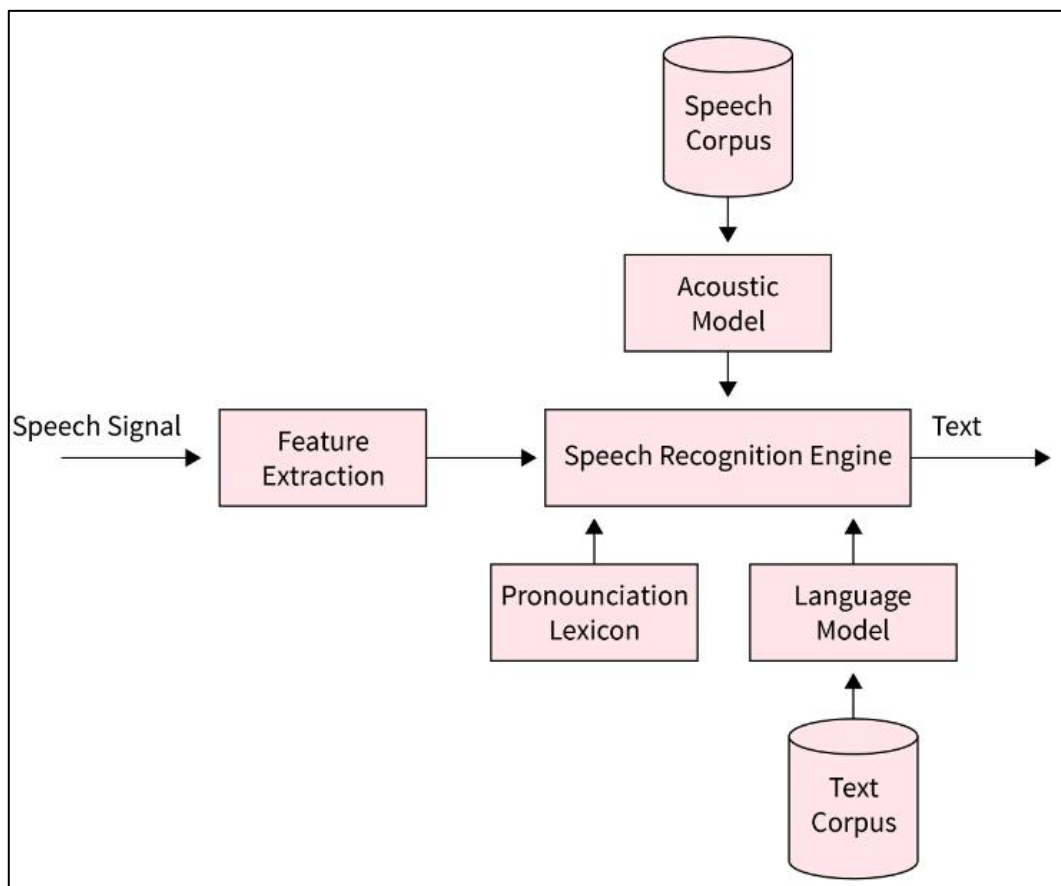
1.4.4. **¿Cómo funciona el Speech Recognition?**

El reconocimiento del habla se basa en dos tecnologías principales: reconocimiento basado en patrones y en modelos estadísticos. El reconocimiento de patrones compara grabaciones de voz con una biblioteca de patrones almacenados. El reconocimiento estadístico

utiliza algoritmos para analizar frecuencia, intensidad y duración del habla, empleando procesamiento del lenguaje natural para adaptarse a patrones individuales y mejorar la precisión. Además, el reconocimiento de voz incorpora técnicas de aprendizaje automático y procesamiento del lenguaje natural para aumentar su exactitud. Figura 4. Se utiliza en asistentes virtuales, dictado, sistemas de control por voz, dispositivos móviles y transcripción de audio, subtítulos y chatbots (pypi.org, 2024).

Figura 4.

Arquitectura del modelo Speech Recognition.



Nota: (Prasanna Kumar, 2023).

1.5. Norma ISO/IEC 25010

En este trabajo el desarrollo de la aplicación de escritorio se aplicará la ISO/IEC 25010 enfocada en la calidad del software, destacando el grado de importancia de la usabilidad como un atributo clave dentro de esta norma (NormasISO.org, 2022).

1.5.1. Definición de usabilidad

Se refiere a la facilidad con la que los usuarios pueden aprender, utilizar y recordarla, así como la satisfacción general del usuario (NormasISO.org, 2022).

1.5.2. Características de usabilidad

Son características específicas de usabilidad establecidas en el estándar, como efectividad, eficiencia, satisfacción del usuario, seguridad, accesibilidad y capacidad de aprendizaje (NormasISO.org, 2022).

1.5.3. Principios de diseño centrados en el usuario

Es la relevancia de implementar principios de diseño enfocados en el usuario con el fin de mejorar la facilidad de uso, tales como la visibilidad del sistema, la retroalimentación informativa y la consistencia (NormasISO.org, 2022).

1.5.4. Métodos de evaluación de usabilidad

Analiza los procedimientos y enfoques habituales para evaluar la facilidad de uso, tales como pruebas de usabilidad, encuestas de satisfacción del usuario, análisis heurísticos y evaluaciones de accesibilidad (NormasISO.org, 2022).

3.1. Trabajos relacionados

Evaluar la idoneidad de la Escala de Usabilidad del Sistema (SUS) en asistentes de voz, proponiendo la Escala de Usabilidad de la Voz (VUS) que considera aspectos únicos de la comunicación por voz (Shah Zwakman et al., 2021). 62 participantes evaluaron la usabilidad de Alexa de Amazon con ambas escalas. El análisis mostró que el SUS tiene limitaciones en medir la usabilidad de asistentes de voz, mientras que la VUS demostró una estructura óptima con tres componentes: usabilidad, afectividad y reconocibilidad/visibilidad. Limitaciones incluyen centrarse en Alexa y el tamaño de la muestra. Además, solo se evaluó la usabilidad, sin considerar privacidad y seguridad.

Evaluar el rendimiento de los sistemas de reconocimiento de voz de inteligencia artificial en conversaciones sociales, explorando su alcance y desafíos en diversos servicios (Barnwal & Gupta, 2022). Mediante una revisión exhaustiva de la literatura, se examinan las aplicaciones del reconocimiento de voz en campos como educación, salud, marketing y servicios empresariales. Los resultados detallan aplicaciones comunes y categorizan la inteligencia artificial. Se destaca el impacto del reconocimiento de voz en diferentes áreas, subrayando su utilidad y aplicabilidad. El estudio señala la falta de precisión y atención en la inteligencia artificial conversacional, abogando por un enfoque más centrado en comprender su valor en conversaciones sociales y diversos contextos de la vida.

Mejorar la precisión del reconocimiento de voz en entornos embebidos mediante un sistema de inteligencia artificial integrado (Zhang & Zhang, 2021). La metodología abarca el diseño hardware, la recopilación y filtrado de características de voz, y la implementación de un modelo de clasificación basado en el algoritmo Naive Bayes mejorado. Las pruebas experimentales en laboratorio revelan un rendimiento destacado, con una tasa de precisión superior al 95% y un uso eficiente de recursos de hardware. A pesar de estos logros, se reconoce la limitación del algoritmo Naive Bayes en algunos entornos de clasificación, afectando la precisión.

Evaluar del desempeño de modelos comerciales de inteligencia artificial (IA) en el reconocimiento de letras de canciones para plataformas de transmisión de audio en internet (Tang, 2022). Se realizaron experimentos con diferentes modelos de IA, incluyendo Watson de IBM, Tencent Cloud de Tencent y I Fly Rec de I Fly Technology, abarcando diversos géneros musicales. El análisis de resultados reveló que el modelo I Fly Rec destacó como el más eficiente. Se identificaron desafíos en el diseño de modelos de reconocimiento de letras, como ruido ambiental, acentos de usuarios y la complejidad en escenarios específicos. Limitaciones incluyen la falta de optimización para reconocimiento melódico y la necesidad de más

experimentos para consolidar conclusiones sobre el rendimiento de los modelos de IA en esta tarea.

Construir un decodificador de transcripción de habla en tiempo real y baja latencia para el reconocimiento automático de habla en transmisiones en vivo (Jorge et al., 2022). Se destaca la integración de modelos acústicos LSTM bidireccionales y modelos de lenguaje interpolados para lograr un rendimiento óptimo. La evaluación empírica abarca conjuntos de datos conocidos como LibriSpeech, TED-LIUM release 2 y RTVE2018, demostrando un rendimiento eficiente. Sin embargo, se identifican limitaciones, como la complejidad computacional, dependencia de conjuntos de datos específicos, limitaciones del modelo de lenguaje y desafíos en la transmisión en vivo, que podrían afectar la aplicabilidad en entornos con recursos limitados y la generalización a diferentes dominios de habla.

Diseñar un sistema de transcripción automática audiovisual, integrando métodos de procesamiento y reconocimiento de voz y visión por computadora (Chaloupka, 2019). La metodología incluye módulos para la segmentación visual, identificación de canales a partir de logotipos, detección de rostros y reconocimiento óptico de caracteres (OCR). Algoritmos específicos se aplican a cada módulo. El sistema logra una tasa de reconocimiento de palabras del 86%, demostrando precisión en la transcripción de contenido de video. Limitaciones incluyen posibles errores en reconocimiento de voz y visión, especialmente en condiciones adversas, y desafíos con palabras poco comunes o idiomas no entrenados en el modelo.

Crear una herramienta que analicen señales de audio y las conviertan en notaciones musicales implica técnicas de procesamiento de señales, reconocimiento de patrones y algoritmos de aprendizaje automático en sistemas de transcripción automática de música (Chis et al., 2018). Estas herramientas facilitan la transcripción automática, beneficiando a investigadores y músicos en la comprensión de grabaciones musicales. No obstante, enfrentan desafíos en la precisión de estilos complejos, la presencia de ruido en las grabaciones y la

incapacidad para capturar expresiones sutiles como el vibrato. Además, la transcripción puede ser difícil con varios instrumentos simultáneos o estructuras musicales complejas.

A-Seeker, según (Affel et al., 2020), presentó una plataforma eficiente de transcripción de audio, permitiendo búsquedas rápidas en archivos de audio y video. Utiliza tecnologías de código abierto, especialmente el motor DeepSpeech, para lograr transcripciones precisas, priorizando usabilidad, asequibilidad y eficiencia. En evaluaciones, A-Seeker logró un WER (Word Error Rate) promedio del 84%, indicando que solo el 16% de las palabras se transcriben incorrectamente. En situaciones comunes como conferencias y podcasts, la precisión alcanzó un impresionante 94%, atribuible a grabaciones de alta calidad y la presencia de un único orador. Presentó limitaciones en tiempo, recursos, acceso a datos y sesgos, afectando la calidad y cantidad de información recolectada, la representatividad de la muestra y la validez de los resultados.

Método para extraer palabras clave como factores de excitación del espectador desde transcripciones de YouTube mediante el análisis estadístico de subtítulos (Hirano et al., 2022). Utiliza el modelo de clasificación binaria Light GBM para etiquetar palabras, y la herramienta LIME para interpretación local y extraer palabras clave. Experimentos demostraron la eficacia del modelo en la base de datos creada. Sin embargo, limitaciones incluyen dependencia de subtítulos para la extracción, restringiendo el método a videos con subtítulos disponibles. Además, la extracción es a nivel de palabra individual, sin considerar el contexto completo de la transcripción, marcando áreas para futuras mejora.

CAPÍTULO 2

2.1. Diseño de lógica del negocio.

El diseño de lógica de negocio establece las reglas, procesos y condiciones que determinan el funcionamiento central de una aplicación. Organiza la gestión de datos, decisiones y cálculos necesarios para cumplir con las reglas del negocio, y se mantiene separado de la interfaz y el acceso a datos para hacer el software más fácil de mantener y expandir (Microsoft, 2023a).

2.1.1. Gestión de grabaciones.

Empieza por capturar el audio que sale y entra por la consola principal de la radio, almacenamiento y organización de las grabaciones de radio se lo hace en formato WAV y guarda con la fecha y hora del inicio de la grabación como nombre del archivo, en una ruta de almacenamiento que se selecciona en las opciones de la aplicación y que sea de fácil acceso para procesos futuros procesos y análisis.

2.1.2. Automatización de las grabaciones de respaldo.

Las grabaciones deben ser automatizadas mediante la opción de la aplicación, los parámetros de automatización son los que se listan en la tabla 3.

Tabla 3.

Parámetros de automatización.

| Nro. | Parámetro |
|------|--|
| 1 | Hora inicio (HH:mm:ss) |
| 2 | Hora fin (HH:mm:ss) |
| 3 | Hora del sistema |
| 4 | Tasa de bits (96, 128,168,192,320 kbps) |
| 5 | Dispositivo de entrada de audio (Menú desplegable) |
| 6 | Días de la semana en checkBox, seleccionables para la repetición de eventos. |
| 7 | Botón activar evento |
| 8 | Ruta de almacenamiento |
| 9 | Prioridad del evento: inmediato y retrasado extendido. |

2.1.3. Búsqueda y análisis de audio.

En esta parte, se utiliza el modelo pre- entrenado Speech Recognition para buscar información dentro del audio, esto hace posterior a la grabación de audio, desde una ventana adicional de la aplicación, donde se manda una palabra o frase como parámetro a buscar en el archivo de audio. Tabla 4.

Tabla 4.

Parámetros de búsqueda y análisis de audio.

| Nro. | Parámetro |
|-------------|--|
| 1 | Archivo .WAV |
| 2 | Palabra clave para buscar. |
| 3 | Ruta de almacenamiento del archivo transcrito. |

2.1.4. Arquitectura de la aplicación de escritorio

Esta Tabla 5 describe el flujo de trabajo, donde las entradas como la consola de audio, archivos, enlaces de transmisión, etc., permiten realizar actividades como la captura y grabación de audio, que resultan en salidas como el archivo de audio en formato WAV y la transcripción en texto, la arquitectura de la aplicación se puede ver a detalle en la Figura 5.

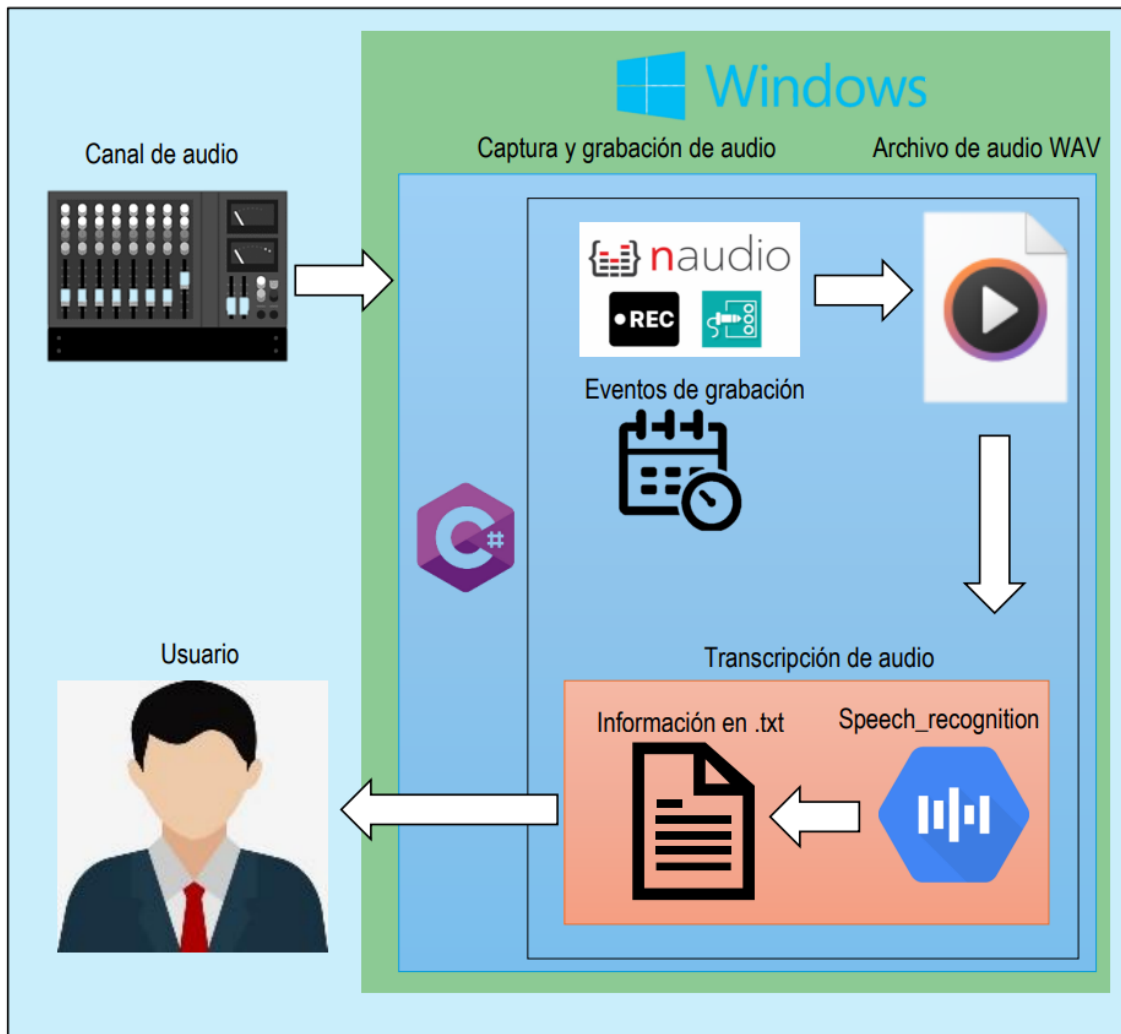
Tabla 5.

Entradas, actividades y salidas de la aplicación.

| Entradas | Actividades | Salidas |
|--|--|---|
| Consola de audio. Archivos de audio. Enlaces de transmisión externa. Hora del sistema. Días de la semana. Ruta del almacenamiento | Captura de audio. Eventos de grabación. Almacenamiento en una ruta de la PC. Transcripción del audio en texto .txt. | Audio grabado en formato WAV. Texto transcrito de audio. Información de audio en texto. |

Figura 5.

Arquitectura de la aplicación de escritorio.



2.2. Configuración del entorno de desarrollo

Herramientas Utilizadas para el desarrollo de esta aplicación son Visual Studio, .NET, C#, NUnit, Microsoft Windows Desktop Runtime, script de Python, modelo Speech Recognition, etc.

2.2.1. Instalación y configuración.

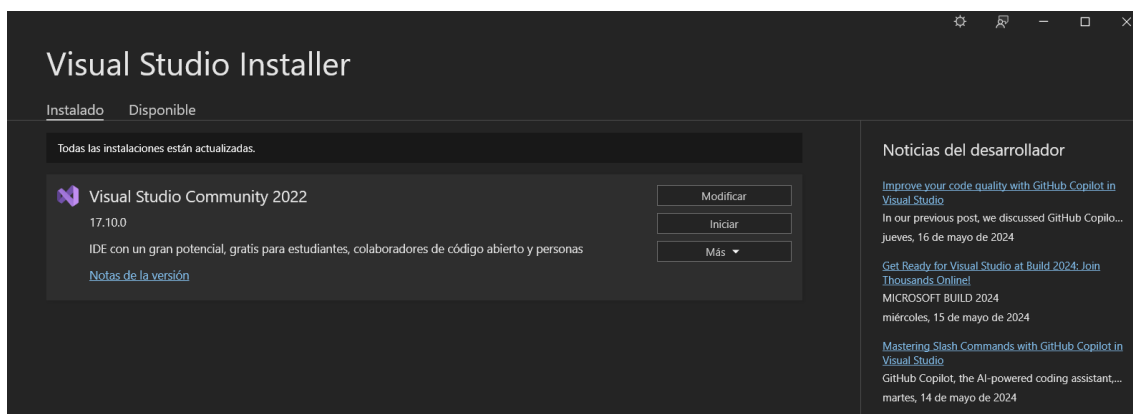
Esta sección describe los pasos esenciales para instalar y configurar el entorno de desarrollo en C#, integrar NUnit para aplicar TDD, NAudio para procesamiento de audio y utilizar el modelo Speech Recognition para el reconocimiento de voz, garantizando el funcionamiento adecuado de cada componente en la aplicación.

2.2.1.1. Entorno de desarrollo.

En la figura 6, se ve la instalación de Visual Studio Community 2022, el IDE que permitirá crear y probar la aplicación en C#. Este entorno proporciona las herramientas necesarias para escribir código, ejecutar pruebas automatizadas, depurar y trabajar con NUnit y el modelo de reconocimiento de voz, facilitando el desarrollo completo de la aplicación (Microsoft, 2024a).

Figura 6.

Instalación de Visual Studio Community 2022.

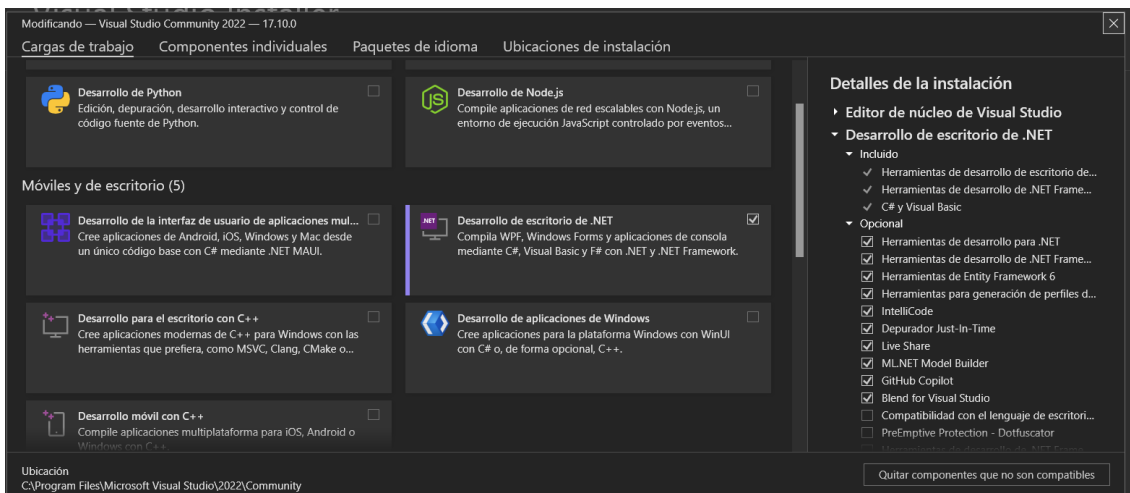


2.2.1.2. Instalación de .NET y C#.

En la figura 7, muestra la instalación de .NET y C#, que proporcionan la plataforma y el lenguaje necesarios para desarrollar y ejecutar la aplicación de escritorio, permitiendo la implementación de su lógica y funcionalidad (Llamas, 2023).

Figura 7.

Instalación de .NET

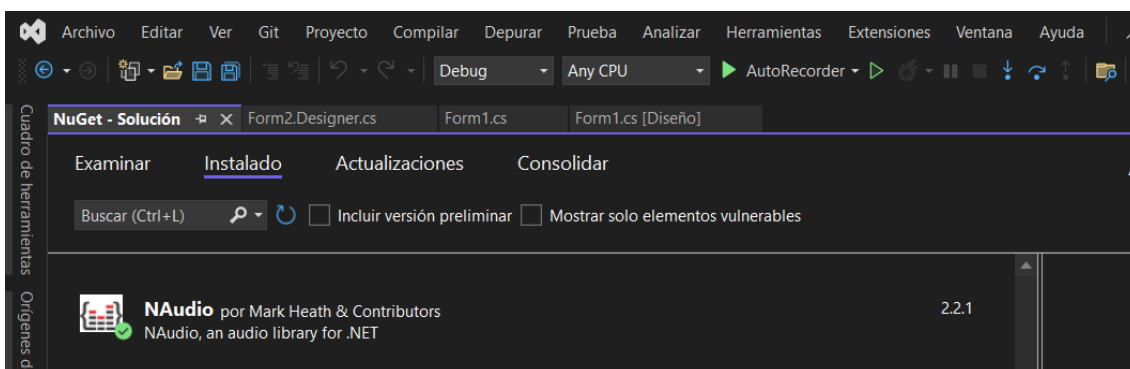


2.2.1.3. Paquete NAudio.

En la figura 8, instala el paquete **NAudio**, que permite integrar funcionalidades para la captura y grabación de audio en la aplicación, facilitando el manejo de sonido en tiempo real (Microsoft, 2023).

Figura 8.

Instalación de paquete NAudio

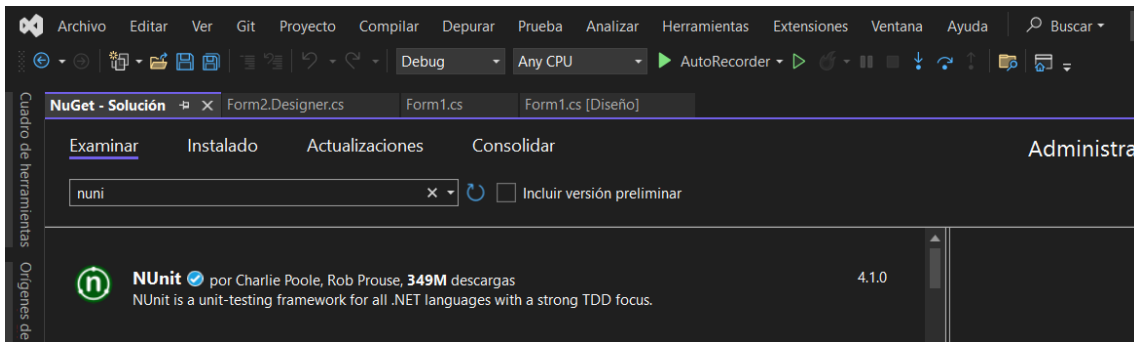


2.2.1.4. Instalación de NUnit.

Instalación de **NUnit**, Figura 9, una herramienta que facilita la implementación de **TDD** y la ejecución de **pruebas unitarias**, garantiza la calidad y el funcionamiento correcto del código durante el desarrollo (*NUnit.Org*, 2023).

Figura 9.

Instalación de NUnit

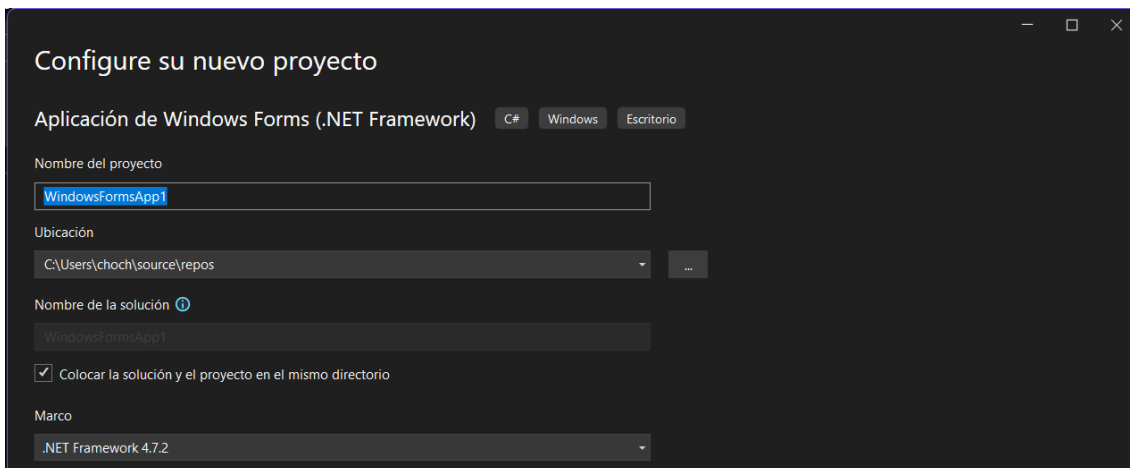


2.2.1.5. Creación del proyecto.

En la Figura 10, está el paso para crear el proyecto y solución C# .NET en el mismo directorio, lo que facilita la organización y gestión de los archivos y recursos del proyecto durante su desarrollo (Microsoft, 2024a).

Figura 10.

Creación del proyecto y la solución C# .NET

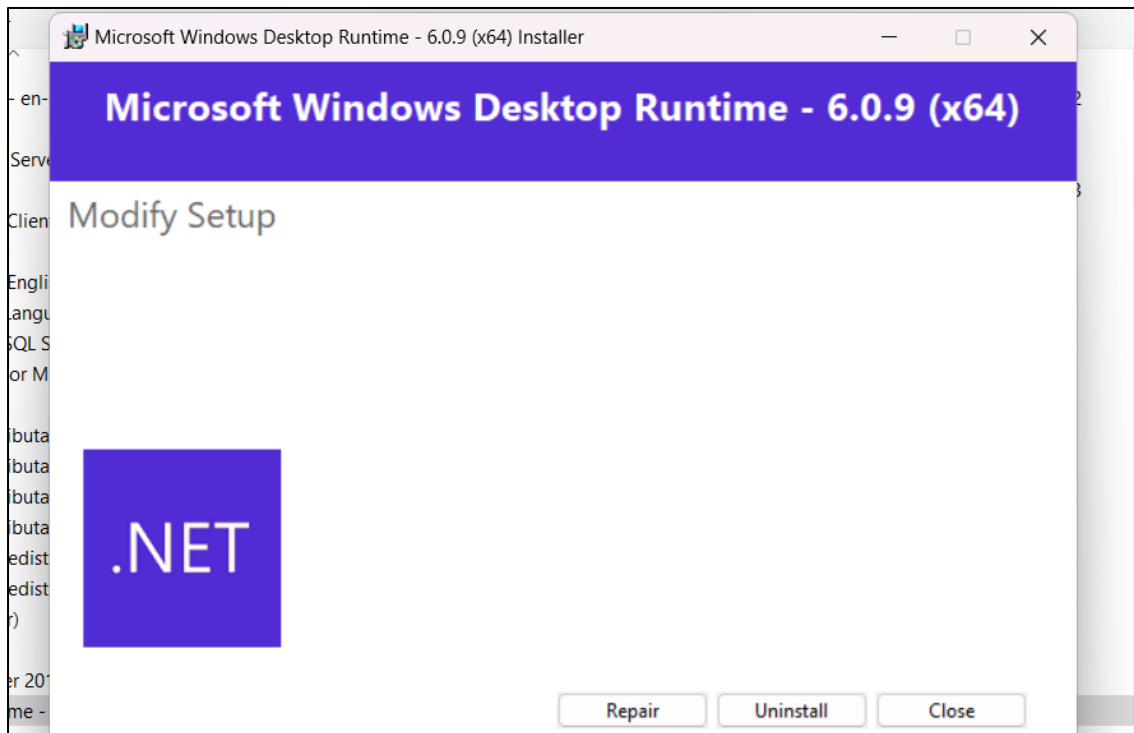


2.2.1.6. Instalación de Microsoft Windows Desktop Runtime.

En la Figura 11 muestra la instalación de Microsoft Windows Desktop Runtime 6.0.9, que es necesario para ejecutar aplicaciones de escritorio .NET en Windows, garantizando su funcionamiento adecuado (Microsoft, 2023b).

Figura 11.

Instalación de Microsoft Windows Desktop Runtime 6.0.9.



2.3. Maquetado de la interfaz de usuario.

Implica diseñar y organizar los elementos visuales de la aplicación, como botones, campos de entrada y menús, para crear una experiencia de usuario intuitiva y eficiente. El objetivo es garantizar que la interfaz sea atractiva, funcional y fácil de usar (IBM, 2024).

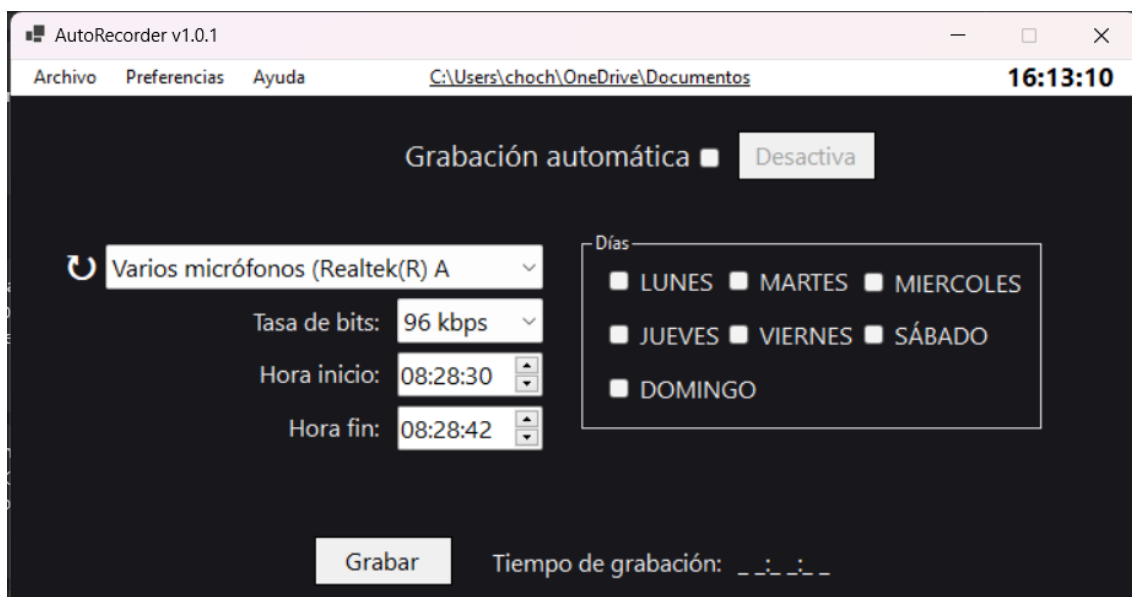
2.3.1. Prototipado de la interfaz de usuario en Windows Forms.

El prototipo de la interfaz de usuario en Windows Forms de la Figura 12, está compuesto por varios elementos clave: campos para ingresar la hora de inicio y hora de fin en formato HH:mm., un campo de solo lectura que muestra la hora del sistema, un menú

desplegable para seleccionar la tasa de bits (96, 128, 168, 192, 320 kbps), otro menú para elegir el dispositivo de entrada de audio, checkBoxes para seleccionar los días de la semana para la repetición de eventos, un botón para activar el evento, un campo para definir la ruta de almacenamiento, y opciones para establecer la prioridad del evento, como inmediato o retrasado. Estos elementos se organizarán de forma clara y funcional para asegurar una experiencia de usuario eficiente e intuitiva (Microsoft, 2024b).

Figura 12.

Prototipo de la interfaz de usuario en Windows Forms.



2.4. Desarrollo de la aplicación de escritorio aplicando TDD y POO

Se han escrito pruebas unitarias con NUnit para cada funcionalidad clave de la aplicación, asegurando así que cualquier cambio en el código no rompa funcionalidades existentes.

A continuación, métodos principales con su respectivo Test e implementación.

2.4.1. Método listar dispositivos test.

Este método de la Figura 13, es una prueba unitaria que verifica que el método *CargarTarjetasDeAudio* cargue correctamente una lista de dispositivos de audio en un

ComboBox. Simula dispositivos de audio y valida que el número de elementos y los nombres en el ComboBox sean correctos.

Figura 13

Método listar dispositivos de entrada disponibles Test.

```
[Test]
public void CargarTarjetasDeAudio_CorrectlyLoadsDevices()
{
    // Mock the device capabilities
    var mockDevices = new List<string> { "Device1", "Device2", "Device3" };
    WaveInEvent.DeviceCount = mockDevices.Count;
    WaveInEvent.GetCapabilities = (device) =>
    {
        return new WaveInCapabilities
        {
            ProductName = mockDevices[device]
        };
    };
    // Act
    _grabacion.CargarTarjetasDeAudio();
    // Assert
    Assert.AreEqual(mockDevices.Count, _comboBoxTargeta.Items.Count);
    for (int i = 0; i < mockDevices.Count; i++)
    {
        Assert.AreEqual(mockDevices[i], _comboBoxTargeta.Items[i].ToString());
    }
}
```

2.4.2. Método listar dispositivos implementación.

El método *CargarTarjetasDeAudio()* de la Figura 14, carga los dispositivos de audio en un ComboBox. Recorre los dispositivos disponibles usando *WaveInEvent.DeviceCount* y obtiene las capacidades de cada uno con *WaveInEvent.GetCapabilities()*. Los nombres de los dispositivos se agregan al *ComboBoxTargeta*, y si existen elementos en el *ComboBox*, selecciona el primero de ellos por defecto.

Figura 14.

Método listar dispositivos de entrada disponibles Implementación.

```
public void CargarTarjetasDeAudio()
{
    for (int device = 0; device < WaveInEvent.DeviceCount; device++)
    {
        var capabilities = WaveInEvent.GetCapabilities(device);
        ComboBoxTargeta.Items.Add(capabilities.ProductName);
    }
    if (ComboBoxTargeta.Items.Count > 0)
    {
        ComboBoxTargeta.SelectedIndex = 0;
    }
}
```

2.4.3. Método activar evento test.

Este método de prueba verifica el correcto funcionamiento del temporizador en la clase Grabación al iniciar y detener la grabación de audio, Figura 15. Configura ciertos parámetros de la interfaz, simula el inicio de la grabación, y luego asegura que el proceso se detenga al alcanzar el tiempo límite, validando así el comportamiento esperado de la grabación a través de comprobaciones en los objetos relacionados.

Figura 15.

Método activar evento Test.

```
[Test]
public void Timer3_Tick_StartsAndStopsRecording()
{
    _grabacion.activarTimer = true;
    _cbActivar.Checked = true;
    _btActivar.Text = "Inactiva";
    _prioridadToolStripMenuItem.Items.AddRange(new object[] { "Alta",
"Media" });
    {
        _prioridadToolStripMenuItem.SelectedIndex = 0;
        _dtHoraInicio.Value = DateTime.Now.AddSeconds(-1);
        _dtHoraFin.Value = DateTime.Now.AddSeconds(1);

        // Act
        _grabacion.Timer3_Tick(null, EventArgs.Empty);

        // Assert - check if recording started
        Assert.IsNotNull(_grabacion.waveIn);

        // Act
        _grabacion.dtHoraFin.Value = DateTime.Now;
        _grabacion.Timer3_Tick(null, EventArgs.Empty);
        // Assert - check if recording stopped
        Assert.IsNull(_grabacion.waveIn);
    }
}
```

2.4.4. Método activar evento implementación.

El método activar evento *Timer3_Tick* de la Figura 16, controla el inicio y la detención de la grabación de audio basado en el tiempo. Primero, verifica si el temporizador y el **CheckBox** de activación están activos. Si la hora actual se encuentra dentro del rango de inicio y fin definidos, y no hay una grabación en curso, inicia la grabación de audio. Si la hora actual supera la hora de fin, detiene la grabación y actualiza el estado del botón para reflejar si la grabación está activa o no. Este método gestiona la grabación de audio automáticamente según el tiempo configurado.

Figura 16.

Método activar evento implementación.

```
public void Timer3_Tick(object sender, EventArgs e)
{
    if (activarTimer && cbActivar.Checked)
    {
        if (DateTime.Now >= dtHoraInicio.Value && DateTime.Now <= dtHoraFin.Value)
        {
            if (waveIn == null)
            {
                waveIn = new WaveIn();
                waveIn.StartRecording();
                btActivar.Text = "Inactiva"; // Cambia el texto del botón a "Inactiva"
            }
        }
        else
        {
            if (waveIn != null)
            {
                waveIn.StopRecording();
                waveIn = null;
                btActivar.Text = "Activa"; // Cambia el texto del botón a "Activa"
            }
        }
    }
}
```

2.4.5. Aplicación de POO en C#

La aplicación se ha desarrollado utilizando principios de POO, lo que facilita la mantenibilidad y escalabilidad del código (Universidad Europea, 2022). Se han utilizado clases y objetos para representar los diferentes componentes del sistema, de esta manera se ha implementado los métodos principales de la aplicación de escritorio, aplicando objetos, clases, métodos y modularidad las que se observa en la Tabla 6.

Tabla 6.

Clases y métodos principales.

| Clases Principales | Métodos |
|---------------------------|--------------------------|
| Grabador | Iniciar grabación |
| | Detener grabación |
| | Guardar grabación |
| Tarjetas | Listar tarjetas de audio |
| | Seleccionar tarjeta |
| Días | Agregar días |

| | |
|--------------|------------------------------|
| | Quitar días |
| | Enviar parámetro |
| | Seleccionar archivo |
| ProcesoAudio | Transcribir audio |
| | Buscar palabra o frase clave |
| | Guardar archivo |
| | Guardar evento |
| Principal | Iniciar evento |
| | Borrar evento |

2.4.6. Integración de Speech Recognition

Speech Recognition se ha integrado en la aplicación para permitir la búsqueda de información en las grabaciones de audio, véase la Figura 18. Se ha utilizado un script de Python para poder implementar la búsqueda de información con el modelo mencionado para ejecutar el modelo de reconocimiento de voz, permitiendo así convertir el audio a texto y realizar búsquedas eficientes.

2.4.7. Script Básico de Python con modelo Speech Recognition

En la figura 17, el script de Python utiliza el modelo pre-entrenado Sepecch Recognition, analiza el audio pregrabado tomando como parámetro la palabra o frase clave para ser analizada y encontrada dentro del audio.

Figura 17.

Script Básico de Python con modelo Speech Recognition.

```
if len(sys.argv) < 3:
    print("Por favor, proporciona una palabra clave y la ruta al archivo de
    audio como argumentos.")
    sys.exit(1)

# Convertir cada segmento de audio a texto utilizando la API de Google y el
idioma español
for segment in segments:
    start_time = float(segment[0]) # Convertir np.float64 a float
    end_time = float(segment[1]) # Convertir np.float64 a float
    audio_chunk = x[int(start_time * Fs):int(end_time * Fs)]
    audio_data = sr.AudioData(audio_chunk.tobytes(), Fs, 2) # 2 bytes para
    audio de 16 bits

    try:
        # Transcribir el segmento de audio
        texto_transcrito = recognizer.recognize_google(audio_data,
        language="es-ES")

        # Mostrar solo si se encuentra la palabra clave
        if palabra_clave.lower() in texto_transcrito.lower():
            tiempo_inicio = segundos_a_tiempo(start_time)
            tiempo_fin = segundos_a_tiempo(end_time)

            # Mostrar texto transcrito y tiempos
            print("_____")
            print(f"_____")
            print(f"{Fore.RED}{texto_transcrito}{Style.RESET_ALL}")
            print(f"Palabra clave encontrada en Tiempo de inicio:
            {tiempo_inicio}, Tiempo de finalización: {tiempo_fin}, Palabra clave:
            {palabra_clave}")
            print("_____")
            print(f"_____")
        else:
            # Transcribe sin anteponer "Texto transcrito" si no se
            encuentra la palabra clave
            print(texto_transcrito)

    except sr.UnknownValueError:
        pass
    except sr.RequestError as e:
        print(f"Error al hacer la solicitud a la API de Google: {e}")
        sys.exit(1)
```

2.4.8. Integración del modelo Speech Recognition.

Este proceso se consigue con el uso de .Net para invocar y mandar el archivo audio mediante la vista del Windows Forms. El proceso de transcripción de audio genera un archivo

.txt, donde se muestra toda la información necesaria para el usuario, en la Figura 18 muestra el modelo de integración.

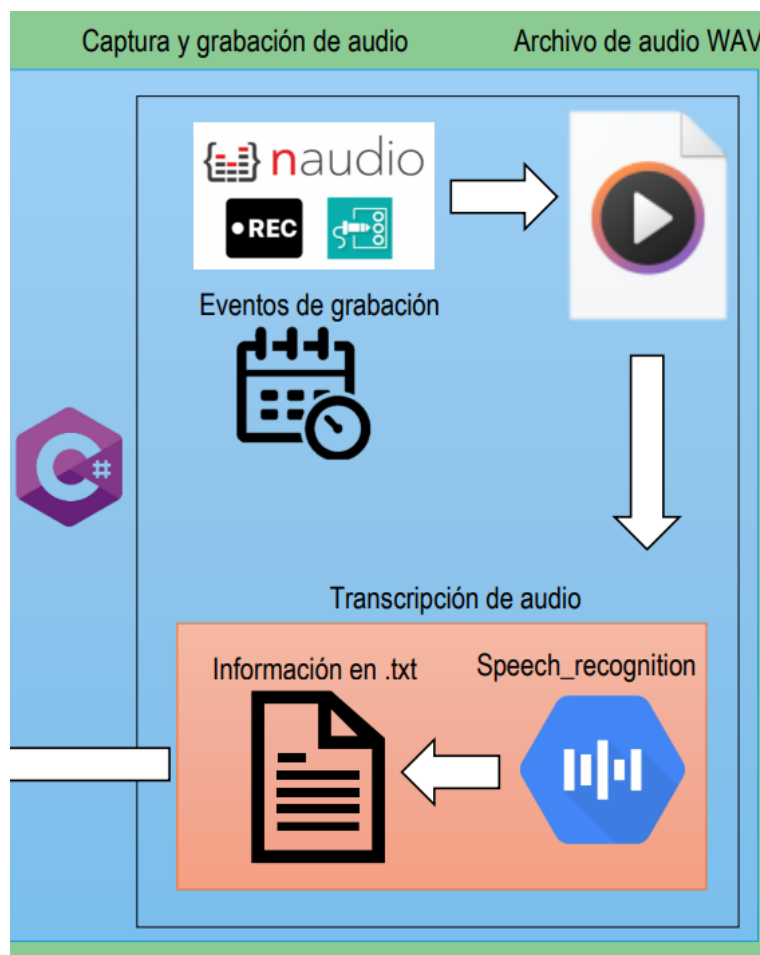
El archivo .txt contiene:

- Audio transcrito.
- Palabra o frase buscada con inicio y fin en hora, minutos y segundos referenciados.
- La palabra buscada se resalta de un color diferente dentro del archivo txt.

Los archivos de audio son segmentados por partes de máximo 1 hora para una mejor precisión en la búsqueda y reconocimiento de palabras dentro del audio.

Figura 18.

Integración del modelo Speech Reconnition.



CAPÍTULO 3

3.2. Metodología de evaluación.

Esta sección describe el proceso y los resultados de la evaluación del rendimiento de la aplicación de escritorio desarrollada para automatizar el respaldo de grabaciones de la programación radial, con un enfoque en la usabilidad según los estándares de calidad de software de la ISO/IEC 25010. La aplicación automatiza el respaldo de grabaciones de programas radiales y utiliza técnicas de inteligencia artificial para la búsqueda de información en el audio. Está diseñada para ser utilizada por operadores y personal de la radio en general.

3.2.1. Criterios de usabilidad.

La evaluación se basará en los siguientes subcaracterísticas de usabilidad que se muestran en la Tabla 7 y son definidos por la ISO/IEC 25010 (NormasISO.org, 2022).

Tabla 7.

Subcaracterísticas e indicadores de usabilidad.

| Subcaracterísticas | Indicador |
|---|---|
| Capacidad para ser comprendido | Integridad de descripción: ¿Qué tan fácil le resultó comprender la interfaz del Software de grabación? Capacidad de demostración: ¿Qué tan intuitiva le pareció la navegación dentro de la aplicación? Funciones evidentes: ¿Qué tan rápido pudo aprender a usar las funciones principales de la aplicación? |
| Capacidad para ser entendido | Efectividad del sistema: ¿Qué tan eficiente encontró la funcionalidad de realizar para realizar grabación? Recuperabilidad de error operacional: ¿Encontró errores o dificultades técnicas mientras usaba la aplicación? Consistencia operacional: ¿Qué tan satisfecho está con el tiempo que toma la aplicación para completar tareas (Configurar la automatización, buscar información)? |
| Protección contra errores de usuario | Verificación de entradas válidas: ¿Encontró errores o dificultades técnicas mientras usaba la aplicación? Prevención del uso incorrecto: ¿Qué tan bien cree que la aplicación protege contra errores del usuario? |
| Estética de la interfaz de usuario | satisfacción del usuario: ¿Cómo calificaría la estética y diseño visual de la aplicación? |
| Accesibilidad | Dado su nivel de importancia mínimo, esta subcaracterística carece de relevancia y no será objeto de evaluación. |
| Extra (recomendable) | Recomendación a otros usuarios: ¿Recomendaría esta aplicación a otras estaciones de radio? |

Nota: (Sifuentes Díaz & Peralta Lujá, 2022).

3.2.2. Instrumentos de evaluación.

- ✓ Pruebas de usabilidad donde se observa a los usuarios interactuar con la aplicación.
- ✓ Encuestas y cuestionarios para obtener feedback de los usuarios: un cuestionario de 10 preguntas que cubre todos los aspectos de usabilidad excepto la subcaracterística de accesibilidad.
- ✓ Se usará la escala de Likert para medir cada pregunta.
- ✓ Escala de puntuación de la encuesta: 1 al 5, donde 1 es mínimo y 5 el máximo.
- ✓ Todas las preguntas tendrán 5 opciones de medición y valoración, excepto la pregunta 7 donde solo tiene 4 opciones.

3.2.3. Población y muestra.

La población se refiere al conjunto completo de individuos que pueden ser objeto de estudio en una investigación, con el fin de obtener una evaluación más completa y precisa de la experiencia de uso, en la Tabla 8 se puede ver la distribución de la población completa.

3.2.3.1. Población objetivo

La población objetivo de la evaluación de usabilidad incluye a todos los posibles usuarios de la aplicación de respaldo de grabaciones de programas radiales. Estos usuarios incluyen locutores, personal administrativo y un técnico de sonido de la estación radial.

3.2.3.2. Selección de la muestra

Para obtener una representación adecuada de la población objetivo, se seleccionó una muestra de 12 individuos, distribuidos de la siguiente manera:

Tabla 8.

Distribución de la población objetivo.

| Locutores | Administrativos | Técnico |
|------------------|------------------------|----------------|
| 8 | 3 | 1 |
| Total: 12 | | |

La muestra fue distribuida anónimamente en rango de edades, sexo, ocupación en la radio y el nivel de conocimiento en informática.

3.2.3.3. Justificación de la distribución de la muestra.

La distribución de la muestra explica cómo se eligieron los participantes y por qué su distribución en diferentes grupos es relevante para el estudio, Figura 9. Su propósito es asegurar que la muestra represente adecuadamente a la población objetivo, permitiendo que los resultados sean representativos y reflejen diversas perspectivas. Esta justificación es clave para demostrar que la selección de la muestra fue adecuada para los objetivos de la investigación y que los hallazgos se pueden generalizar a toda la población.

Tabla 9.

Justificación de la Distribución de la Muestra.

| Distribución | Justificación | Representatividad |
|---|---|--------------------------|
| Locutores (8 individuos) | Grupo más grande, representa a la mayoría de los usuarios finales que interactuarán directamente con la aplicación. Su experiencia y retroalimentación son fundamentales para evaluar la facilidad de uso y la eficiencia en la realización de copias de seguridad de audio (Unila, 2022). | 66.67% |
| Administrativos (3 individuos) | Grupo considerable, representa una pequeña parte de los usuarios finales que interactúan indirectamente con la aplicación. Evaluar su experiencia es esencial para garantizar que la aplicación sea accesible y fácil de usar para todos los miembros del equipo, independientemente de su nivel de interacción con ella (Unila, 2022). | 25% |
| Técnico (1 individuo) | Grupo pequeño, los técnicos de sonido poseen un conocimiento técnico más avanzado y pueden ofrecer observaciones valiosas sobre el desempeño técnico de la aplicación, la eficiencia en la búsqueda de información dentro del audio y la solidez del sistema en general (Unila, 2022). | 8.33% |
| | | 100% |

La selección de esta muestra se hizo para asegurarse de que todas las perspectivas relevantes fueran consideradas al evaluar la usabilidad de la aplicación. La distribución justa permite que la aplicación funcione bien para todos los tipos de usuarios al identificar problemas y áreas de mejora desde diferentes puntos de vista.

3.3. Procedimiento de Evaluación.

Define los pasos para llevar a cabo una evaluación de usabilidad, desde la planificación hasta el análisis de los resultados. Incluye la definición de objetivos, selección de usuarios, pruebas de usabilidad, recopilación de datos y análisis de la experiencia del usuario, con el fin de identificar mejoras en el sistema evaluado.

3.3.1. Plan de pruebas.

Las pruebas de usabilidad se realizarán en un entorno controlado. Los participantes completarán una serie de tareas específicas usando la aplicación:

- ✓ Configurar la automatización de un respaldo de grabación.
- ✓ Buscar información en una grabación usando el apartado de inteligencia artificial.
- ✓ Recuperar un respaldo previamente realizado.

3.3.2. Recolección de datos.

Los datos se recopilarán mediante:

- ✓ Grabaciones de las sesiones de prueba.
- ✓ Notas de observación tomadas en la evaluación.
- ✓ Respuestas a encuestas y cuestionarios.
- ✓ Comentarios obtenidos durante las entrevistas.

3.3.3. Análisis de datos.

Se utilizaron métodos cuantitativos y cualitativos para analizar los datos recopilados, como el análisis de frecuencia de errores, tiempos de tarea y satisfacción del usuario. Resultados de la Evaluación.

3.3.4. Resultados de encuesta.

A continuación, muestra los resultados obtenidos de cada pregunta con su respectiva escala, ponderación y número de respuesta por opción.

Fórmula:

$$\text{Total, Puntuación} = \text{ponderación} * \text{número de respuesta}$$

Donde la **Ponderación total es igual** = número de opciones * muestra total

Muestra total = 12

$$\text{Puntuación en \%} = \frac{\text{Total, Puntuación}}{\text{Ponderación Total}} * 100\%$$

Pregunta 1: relacionada con la **Capacidad para ser comprendido** en la norma ISO/IEC 25010, dentro de la usabilidad, Tabla 10, y se refiere a lo fácil que es para los usuarios entender y utilizar el software.

Tabla 10.

Comprensión de la interfaz del Software de grabación.

| ¿Qué tan fácil le resultó comprender la interfaz del Software de grabación? | | | | |
|---|----------|----------|----------|-----------|
| Ponderación | | | | |
| 1 | 2 | 3 | 4 | 5 |
| Muy difícil | Difícil | Neutral | Fácil | Muy fácil |
| Número de respuesta | | | | |
| 0 | 1 | 3 | 6 | 2 |
| Total, Puntuación: 45/60 = 75% | | | | |

Pregunta 2: la pregunta de la Tabla 11, está relacionada con la subcaracterística de eficiencia de aprendizaje en usabilidad, que evalúa la rapidez con la que un usuario puede aprender a utilizar las funciones principales de la aplicación.

Tabla 11.

Uso de las funciones principales del Software de grabación.

| ¿Qué tan rápido pudo aprender a usar las funciones principales de la aplicación? |
|--|
|--|

| Ponderación | | | | |
|---------------------------------------|----------|----------|----------|------------|
| 1 | 2 | 3 | 4 | 5 |
| Muy lento | lento | Neutral | Rápido | Muy rápido |
| Numero de respuesta | | | | |
| 0 | 1 | 3 | 5 | 3 |
| Total, Puntuación: 46/60 = 76% | | | | |

Pregunta 3: se refiere a la facilidad con que los usuarios pueden acceder y moverse entre las funciones del programa, Tabla 12. Un diseño intuitivo permite a los usuarios encontrar rápidamente las opciones necesarias sin dificultad, lo que se relaciona con la eficiencia de la operación y facilidad de uso.

Tabla 12.

Navegación dentro Software de grabación.

| Ponderación | | | | |
|---------------------------------------|----------|----------|-----------|---------------|
| 1 | 2 | 3 | 4 | 5 |
| Muy confusa | Confusa | Neutral | Intuitiva | Muy intuitiva |
| Numero de respuesta | | | | |
| 0 | 0 | 3 | 7 | 2 |
| Total, Puntuación: 57/60 = 95% | | | | |

Pregunta 4: Tabla 13, se refiere a la capacidad del software para realizar copias de seguridad de audio de manera rápida y con un uso mínimo de recursos, sin errores. Está relacionada con la eficiencia de rendimiento.

Tabla 13.

Eficiencia de la funcionalidad de realizar respaldos de audio.

| Ponderación | | | | |
|---------------------------------------|-------------|----------|-----------|---------------|
| 1 | 2 | 3 | 4 | 5 |
| Muy ineficiente | Ineficiente | Neutral | Eficiente | Muy eficiente |
| Puntuación | | | | |
| 0 | 0 | 1 | 7 | 4 |
| Total, Puntuación: 53/60 = 88% | | | | |

Pregunta 5: la Tabla 5, se refiere a la capacidad del software para localizar datos específicos dentro de los archivos de audio. Está relacionada con la funcionalidad y la usabilidad.

Tabla 14.

Búsqueda de información en el audio de respaldo.

| ¿Qué tan claro y accesible encontró la función de búsqueda de información en el audio? | | | | |
|--|-----------------|----------|---------------|-------------------|
| Ponderación | | | | |
| 1 | 2 | 3 | 4 | 5 |
| Muy difícil de usar | Difícil de usar | Neutral | Fácil de usar | Muy fácil de usar |
| Numero de respuesta | | | | |
| 0 | 0 | 2 | 7 | 3 |
| Total, Puntuación: 49/60 = 81% | | | | |

Pregunta 6: la Tabla 15, está relacionado con la eficiencia de rendimiento en la norma ISO/IEC 25010.

Tabla 15.

Tiempo para completar tareas.

| ¿Qué tan satisfecho está con el tiempo que toma la aplicación para completar tareas (Configurar la automatización, buscar información)? | | | | |
|---|--------------|----------|------------|----------------|
| Ponderación | | | | |
| 1 | 2 | 3 | 4 | 5 |
| Muy insatisfecho | Insatisfecho | Neutral | Satisfecho | Muy satisfecho |
| Numero de respuesta | | | | |
| 0 | 0 | 0 | 8 | 4 |
| Total, Puntuación: 52/60 = 86.7 | | | | |

Pregunta 7: los errores o dificultades técnicas en el software de grabación se refieren a fallos o problemas durante su uso. La Tabla 17, está relacionado con la fiabilidad según la norma ISO/IEC 25010, que evalúa su capacidad para operar sin interrupciones.

Tabla 16.*Errores o dificultades técnicas mientras usaba el Software de grabación.*

| ¿Encontró errores o dificultades técnicas mientras usaba la aplicación? | | | | |
|---|---------------|-----------|-------|--|
| Ponderación | | | | |
| 1 | 2 | 3 | 4 | |
| Frecuentemente | Algunas veces | Raramente | Nunca | |
| Numero de respuesta | | | | |
| 0 | 0 | 5 | 7 | |
| Total, Puntuación: 43/48 = 89% | | | | |

Pregunta 8: la estética y diseño visual del software se refiere a la apariencia y disposición de su interfaz, la tabla 17, está relacionado con la usabilidad y atributos de interfaz.

Tabla 17.*Estética y diseño visual del Software de grabación.*

| ¿Cómo calificaría la estética y diseño visual de la aplicación? | | | | |
|---|------|---------|-------|-----------|
| Ponderación | | | | |
| 1 | 2 | 3 | 4 | 5 |
| Muy malo | Malo | Neutral | Bueno | Muy bueno |
| Numero de respuesta | | | | |
| 0 | 0 | 2 | 8 | 2 |
| Total, Puntuación: 48/60 | | | | |

Pregunta 9: la resiliencia a fallos es la capacidad del software para seguir funcionando correctamente frente a errores imprevistos, la Tabla 18, está relacionado con la fiabilidad.

Tabla 18.*Resiliencia a fallos.*

| Pregunta 9: ¿Qué tan bien cree que la aplicación protege contra errores del usuario? | | | | |
|--|-----|---------|------|----------|
| Ponderación | | | | |
| 1 | 2 | 3 | 4 | 5 |
| Muy mal | Mal | Neutral | Bien | Muy bien |
| Numero de respuesta | | | | |
| 0 | 0 | 0 | 6 | 6 |
| Total, Puntuación: 54/60 = 90% | | | | |

Pregunta 10: Tabla 19, mide la satisfacción general del usuario, reflejando la usabilidad y funcionalidad del software según la norma ISO/IEC 25010.

Tabla 19.

Recomendación a otros usuarios.

| Pregunta 10: ¿Recomendaría esta aplicación a otras estaciones de radio? | | | | |
|--|------------------|-----------------|------------------|--------------------|
| Ponderación | | | | |
| 1 | 2 | 3 | 4 | 5 |
| Definitivamente no | Probablemente no | No estoy seguro | Probablemente sí | Definitivamente sí |
| Numero de respuesta | | | | |
| 0 | 0 | 0 | 6 | 6 |
| Total, Puntuación: 54/60 = 90% | | | | |

3.3.4.1. Puntuación global de usabilidad.

La puntuación global de usabilidad se basa en la evaluación general del software basada en su facilidad de uso, eficiencia y satisfacción de todos los usuarios, reflejando su desempeño en términos de usabilidad según la norma ISO/IEC 25010. Para este calculo se emplea la siguiente formula.

$$\begin{aligned}
 & \textit{Puntuacion global} \\
 & = \frac{\textit{Total puntuación pregunta 1} + \dots + \textit{Total puntuación pregunta 10}}{\textit{Ponderacion global Total}} \\
 & * 100\%
 \end{aligned}$$

Donde la **Ponderación global Total** es igual a la suma de todas las ponderaciones por pregunta que es igual 588.

$$\textit{Puntuacion global} = \frac{45 + 46 + 57 + 53 + 49 + 52 + 43 + 48 + 54 + 54}{588} * 100\%$$

$$\textit{Puntuacion global} = \frac{501}{588} * 100\%$$

$$\textit{Puntuacion global} = 85.2\%$$

3.3.4.2. Criterio de evaluación para la característica de usabilidad.

Según el criterio de evaluación de usabilidad ISO/IEC 25010, Figura 20, se ha definido que entre 80 a 10 se considera Alta calidad (Sifuentes Díaz & Peralta Lujá, 2022). En la evaluación de la aplicación de grabación se obtuvo un puntaje del **85.2** obteniendo una alta calidad.

Tabla 20.

Criterio de evaluación para la característica de usabilidad.

| Rango de puntuación | Criterio de evaluación para la característica de usabilidad. |
|---------------------|--|
| 100 - 80 | Alta calidad |
| 79 - 50 | Buena calidad |
| 49 - 20 | Regular calidad |
| 19 - 0 | Mala calidad |

3.3.4.3. Observaciones de las pruebas de usabilidad

De acuerdo con las pruebas realizadas se encontraron las siguientes observaciones

- Problemas recurrentes en la navegación.
- Interfaz moderadamente intuitiva.
- Funciones que resultaron confusas para los usuarios.
- Áreas donde los usuarios necesitaron asistencia.
- Áreas donde las funciones no eran evidentes.
- Mensajes y alertas no muy detalladas.

3.3.5. Análisis y discusión.

La evaluación de la usabilidad de la aplicación basado en los datos recogidos y los criterios de la ISO/IEC 25010 dieron los siguientes criterios, y se muestra en la Tabla 21.

Tabla 21.

Análisis y aceptación de Usabilidad

| Subcaracterística | Análisis |
|--|--|
| Capacidad para ser comprendido | La mayoría de los usuarios encontró la interfaz comprensible, pero hubo algunas dificultades iniciales. |
| Capacidad para ser entendido | Los usuarios pudieron aprender a usar las funciones principales rápidamente. |
| Protección contra errores de usuario | Algunos usuarios encontraron problemas con la navegación y la recuperación de los respaldos. |
| Estética de la interfaz de usuario | Los usuarios mostraron interés en la aplicación, pero sugirieron mejoras en el diseño de la interfaz. |
| Capacidad de protección contra errores: | Se identificaron varios errores comunes que los usuarios cometieron. Pero la aplicación se mantuvo estable. |
| Accesibilidad | Dado su nivel de importancia mínimo, esta subcaracterística carece de relevancia y no será objeto de evaluación. |

3.3.5.1. Identificación de Problemas

- Problemas de navegación y localización de funciones.
- Errores en la configuración de automatización.
- Falta de indicaciones claras en ciertas áreas de la interfaz.
- Confusión en el modo de configuración (prioridad)

3.3.5.2. Propuestas de Mejora

- Simplificar la navegación y hacer más visibles las funciones principales.
- Proporcionar indicaciones y ayudas más claras en la interfaz.
- Simplificar los pasos de configuración para que sea más sencillo y utilizar menos clics.
- Etiquetas más claras y directas.
- Mejorar los mensajes de las alertas para mayor comprensión.

3.4. Conclusiones

La aplicación de la metodología TDD (Desarrollo Guiado por Pruebas) con NUnit en el desarrollo del proyecto ha contribuido a un código más robusto y bien estructurado, mejorando la calidad del software y facilitando el mantenimiento y la evolución futura de la aplicación.

La utilización del modelo pre-entrenado de *Speech Recognition* permitió obtener una precisión aceptable en la transcripción de audio a texto, aunque no alcanzó un nivel de exactitud alto. Aun así, fue suficiente para facilitar la búsqueda y recuperación de información específica dentro de las grabaciones, cumpliendo con los requisitos mínimos del sistema.

La obtención de dispositivos de audio que garantizaran una buena calidad de captura fue compleja debido a la necesidad de compatibilidad entre el hardware y el software del sistema. Fue esencial que los dispositivos de audio funcionaran bien con los controladores y las bibliotecas en C#, y que fueran compatibles con los algoritmos de inteligencia artificial para el análisis de audio. Cualquier incompatibilidad podía afectar la precisión del sistema, lo que hizo necesario realizar configuraciones específicas y pruebas exhaustivas. Esta interacción entre hardware y software añadió una capa importante de complejidad al desarrollo de la aplicación.

La evaluación del rendimiento de la aplicación bajo el marco de la norma ISO/IEC 25010 ha mostrado que la aplicación cumple con los criterios de usabilidad, siendo intuitiva y fácil de usar para los usuarios finales, lo cual es esencial para la adopción y utilización efectiva de la herramienta.

3.5. Recomendaciones

A pesar de que la aplicación cumple con los criterios básicos de usabilidad, se recomienda realizar pruebas de usuario adicionales para identificar áreas de mejora en la

interfaz, buscando optimizar la experiencia del usuario y facilitar aún más la navegación y el uso de la aplicación.

Para mantener la precisión en la transcripción de audio, se sugiere actualizar periódicamente el modelo de Speech Recognition y considerar entrenarlo con datos específicos de la programación radial para mejorar su capacidad de reconocimiento en ese contexto específico.

Se recomienda explorar servicios en la nube para el almacenamiento, lo que podría ofrecer mayor escalabilidad, seguridad y accesibilidad a los datos respaldados o utilizar dispositivos de almacenamiento externo.

Bibliografía

- Carrasco, D. (3 de enero de 2023). *Herramientas de IA con las que puedes convertir de audio a texto*. Obtenido de <https://marketing4ecommerce.net/top-10-herramientas-de-ia-con-las-que-puedes-convertir-audios-en-textos/>
- CORDICOM. (15 de octubre de 2014). *Suplemento del Registro Oficial No. 354*. Quito, Pichincha, Ecuador. Obtenido de <https://www.consejodecomunicacion.gob.ec/wp-content/uploads/downloads/2021/01/lotaip/REGLAMENTO%20QUE%20ESTABLECE%20LOS%20PAR%20C3%81METROS%20T%20C3%89CNICOS%20PARA%20LA%20DEFINICI%20N%20DE%20AUDIENCIAS,%20FRANJAS%20HORARIA%20S.pdf>
- De Cervantes Saavedra, M. (2021). *Codster*. Obtenido de <https://codster.io/blog/desarrollo-de-aplicaciones-web-vs-escritorio/>
- Euroinnova. (2023). *Euroinnova*. Obtenido de <https://www.euroinnova.ec/blog/aplicaciones-de-escritorio>
- Iriz González, J. (2019). *Procesamiento de Audio con Técnicas de Inteligencia Artificial*. Tesis. Obtenido de https://e-archivo.uc3m.es/bitstream/handle/10016/29348/TFG_Jesus_Iriz_Gonzalez.pdf?sequence=1&isAllowed=y
- Lema, S., & Romero, P. (junio de 28 de 2023). *Ley Orgánica de Comunicación en Ecuador. Génesis, conflicto, reforma y perspectivas*. Universidad Politécnica Salesiana. Quito: Editorial Universitaria Abya-Yala. Obtenido de <http://dspace.ups.edu.ec/handle/123456789/25071>
- Llamas, L. (22 de abril de 2023). *Cómo procesar ficheros de audio en .NET con NAudio*. Obtenido de <https://www.luisllamas.es/naudio/>

- Maida, E., & Pacienza, J. (2015). *Metodologías de desarrollo de software*. Tesis, Universidad de Argentina. Obtenido de <https://repositorio.uca.edu.ar/bitstream/123456789/522/1/metodologias-desarrollo-software.pdf>
- Martín Pena, D., & Parejo Cuellar, M. (2018). *Las Tecnologías de la Información y la Comunicación en las radios universitarias españolas en el periodo 2012-2016*. Artículo. Obtenido de <https://www.scielo.br/j/tinf/a/vqJP5tzRhbNkCJXYqkJWXDm/?format=pdf>
- Microsoft. (2023). *Naudio*. Obtenido de <https://learn.microsoft.com/es-es/nuget/>
- Morales, M. (26 de septiembre de 2023). Importancia de los respaldos de audio en la radio. (D. Chocho, Entrevistador) Otavalo, Imbabaura, Ecuador.
- Mosquera Aguilar, J. (2022). *Adaptación digital de la radio y la captación de auspiciantes; Caso: Radio Jubones*. Universidad de Guayaquil. Guayaquil: Universidad de Guayaquil: Facultad de Comunicación Social. Obtenido de <http://repositorio.ug.edu.ec/handle/redug/69050>
- Parra, D., & Ramírez, J. (2018). *Diseño, desarrollo e implementación de software de escritorio y aplicativo móvil para la administración y gestión de venta y preventa de la distribuidora Buitrago*. Tesis, Universidad Piloto de Colombia. Obtenido de <http://repository.unipiloto.edu.co/handle/20.500.12277/6111>
- Perugachi, R. (28 de septiembre de 2023). LEY ORGANICA DE COMUNICACION. (D. Chocho, Entrevistador) Otavalo, Imbabura, Ecuador.
- Radios de América. (2022). La radio en las TIC. Obtenido de <http://titoballesteros.blogspot.com/2011/09/la-radio-en-las-tic.html>

Sifuentes Díaz, Y. M., & Peralta Lujá, J. L. (2022). *Modelo de medición y evaluación de calidad del software basado en la norma ISO/IEC 25000 para .* Trujillo: TecnoHumanismo. Revista Científica.

TECH Education. (2023). *TECH Education.* Obtenido de <https://www.techtitute.com/ec/informatica/especializacion/especializacion-desarrollo-software-aplicaciones-escritorio#:~:text=El%20desarrollo%20de%20software%20para%20aplicaciones%20de%20escritorio%20es%20un,funcionalidades%20espec%C3%ADficas%20para%20l os>

Acosta, A. J. (2017). *Impacto de la implementación del sistema de control y monitoreo de incidencias utilizando C#.Net para la empresa Security Fast, Tacna, 2017.*

Boullosa, A. (2014). *Speech Recognition, accesibilidad para la web Desarrollo Interfaces por Voz.* <https://www.dreams.es/transformacion-digital/desarrollo-interfaces-por-voz/speech-recognition-accesibilidad-para-la-web>

Corredor, T., Iván, O., Valdez, N., & Rolando, E. (2017). *Aplicación y la evaluación de la metodología desarrollo orientado por pruebas (TDD), caso de estudio: spot.co.*

Evin, D. (2019). *Tecnologías del Habla Reconocimiento Automático del Habla.*

Fermin González, D., Asiel, R., & Faez, R. (2015). *Desarrollo de pruebas unitarias automatizadas a los módulos de seguridad y gestión de medias del sistema XILEMA PRIMICIA 2.0.*

Flores, H. (2018). *Universidad Nacion al Hermilio Valdizán escuela de post grado.*

Ibarra, O. F., & Hurtado, S. J. (2022). *Aplicación web usando el framework modular multiplataforma .NET CORE para el refuerzo académico de la educación general básica*

- virtual en la unidad educativa “González Suárez” de la ciudad de Ambato.*
<https://repositorio.uta.edu.ec:8443/jspui/handle/123456789/34829>
- IBM. (2024, October 7). *Interfaces de usuario - Documentación de IBM.*
<https://www.ibm.com/docs/es/i/7.5?topic=applications-user-interfaces>
- Jerez, D. S. (2017). *Análisis de métodos, técnicas y herramientas de verificación y validación de software, aplicados en la dirección de tecnología de información y comunicación de la Universidad Técnica de Ambato.*
- Macas, C., David, R., Cepeda, R., & Mauricio, A. (2021). *Universidad Cacional de Chimborazo Facultad de Ingeniería.*
- Microsoft. (2023a, April 6). *Implementar la lógica de negocios (LINQ to SQL) - ADO.NET | Microsoft Learn.* <https://learn.microsoft.com/es-es/dotnet/framework/data/adonet/sql/linq/implementing-business-logic-linq-to-sql>
- Microsoft. (2023b, October 12). *Aplicaciones de la Plataforma universal de Windows, Windows Runtime y tiempo de ejecución de C | Microsoft Learn.*
<https://learn.microsoft.com/es-es/cpp/c-runtime-library/windows-store-apps-the-windows-runtime-and-the-c-run-time?view=msvc-170>
- Microsoft. (2024a, June 21). *¿Qué es el IDE de Visual Studio? | Microsoft Learn.*
<https://learn.microsoft.com/es-es/visualstudio/get-started/visual-studio-ide?view=vs-2022>
- Microsoft. (2024b, November 5). *Qué es Windows Forms - Windows Forms .NET | Microsoft Learn.* <https://learn.microsoft.com/es-es/dotnet/desktop/winforms/overview/?view=netdesktop-9.0>
- NormasISO.org. (2022). *► ISO 25010: Mejora calidad y satisfacción del usuario en software.*
<https://normasiso.org/norma-iso-25010/>
- NUnit.org. (2023). NUnit.Org. <https://nunit.org/>

- Pastor, F. (2022). *Análisis de audio mediante técnicas de aprendizaje profundo*.
<https://riunet.upv.es/handle/10251/187795>
- Paucar, A., & Eduardo, F. (2017). *Desarrollo de pruebas unitarias automáticas Para el sistema web escolástico del instituto Tecnológico Superior Stanford*”.
- Prasanna Kumar. (2023). *Speech Recognition Architecture - Scaler Topics*.
<https://www.scaler.com/topics/nlp/architecture-of-automatic-speech-recognition/>
- pypi.org. (2024). *SpeechRecognition · PyPI*. <https://pypi.org/project/SpeechRecognition/>
- SHIFT ASIA. (2024, June 20). *Exploring Test-Driven Development (TDD): A Comprehensive Overview*.
<https://blog.shiftasia.com/exploring-test-driven-development-tdd-a-comprehensive-overview/>
- Unila. (2022). *¿Qué se Necesita para Trabajar en la Radio? | UNILA*.
<https://www.unila.edu.mx/que-se-necesita-para-trabajar-en-la-radio/>
- Universidad Europea. (2022). *Programación orientada a objetos | Blog UE*.
<https://universidadeuropea.com/blog/programacion-orientada-objetos/>
- Veintimilla, A., & Cuenca, L. (2014). *Departamento de ciencias de la computación tema: estudio de la técnica test driven development (TDD) y desarrollo del sistema para la administración de consultorios médicos*.

Anexos

Anexo 1. Entrevista de la situación actual.

Preguntas al técnico y al director de la radio sobre los recursos para la grabación.

1. Hardware:

Productividad: ¿En qué porcentaje el hardware actual permite completar las grabaciones de manera eficiente y sin retrasos?

Usabilidad: ¿Qué porcentaje de los usuarios considera fácil de usar el hardware de grabación, en una escala del 0% (muy difícil) al 100% (muy fácil)?

Disponibilidad: ¿En qué porcentaje del tiempo programado el hardware está disponible para su uso sin interrupciones?

Confiabilidad: ¿Qué porcentaje de las veces el hardware funciona correctamente sin fallos durante las grabaciones?

Satisfacción: ¿Qué porcentaje de los usuarios está satisfecho con el rendimiento del hardware durante las grabaciones?

2. Software:

Productividad: ¿En qué porcentaje el software actual permite realizar las grabaciones sin errores o problemas técnicos?

Usabilidad: ¿Qué porcentaje de los usuarios considera fácil de usar el software de grabación, en una escala del 0% (muy difícil) al 100% (muy fácil)?

Disponibilidad: ¿Qué porcentaje del tiempo el software está disponible para su uso sin problemas técnicos?

Confiabilidad: ¿En qué porcentaje de las veces el software funciona sin fallos o errores durante las grabaciones?

Satisfacción: ¿Qué porcentaje de los usuarios está satisfecho con el rendimiento del software durante las grabaciones?

3. Archivo de respaldo:

Productividad: ¿Qué porcentaje de las veces el sistema de respaldo permite realizar copias de seguridad exitosamente?

Usabilidad: ¿Qué porcentaje de los usuarios considera fácil utilizar el sistema de respaldo para almacenar las grabaciones?

Disponibilidad: ¿Qué porcentaje del tiempo el sistema de respaldo está disponible para su uso sin problemas?

Confiabilidad: ¿En qué porcentaje de las veces el archivo de respaldo guarda correctamente las grabaciones sin pérdida de datos?

Satisfacción: ¿Qué porcentaje de los usuarios está satisfecho con el proceso y funcionamiento del archivo de respaldo?

Preguntas al técnico y al director de la radio sobre la actividad humana para la grabación

1. Iniciar grabación:

Productividad: ¿En qué porcentaje de las ocasiones se logra iniciar la grabación correctamente y sin retrasos?

Usabilidad: ¿Qué porcentaje de los usuarios considera fácil el proceso de iniciar la grabación, en una escala del 0% (muy difícil) al 100% (muy fácil)?

Disponibilidad: ¿Qué porcentaje de las veces el equipo está listo y disponible para iniciar una grabación sin problemas?

Confiabilidad: ¿Qué porcentaje de las veces el inicio de la grabación ocurre sin errores o fallos?

Satisfacción: ¿Qué porcentaje de los usuarios está satisfecho con el proceso de inicio de la grabación?

2. Finalizar grabación:

Productividad: ¿Qué porcentaje de las grabaciones se finalizan correctamente en el tiempo previsto?

Usabilidad: ¿Qué porcentaje de los usuarios encuentra fácil finalizar una grabación, en una escala del 0% al 100%?

Disponibilidad: ¿Qué porcentaje de las veces el sistema está disponible para finalizar la grabación sin fallos?

Confiabilidad: ¿Qué porcentaje de las veces el sistema finaliza la grabación sin errores?

Satisfacción: ¿Qué porcentaje de los usuarios está satisfecho con el proceso de finalizar una grabación?

3. Automatización (sin datos):

Productividad: Actualmente, ¿qué porcentaje de las tareas de grabación están automatizadas?

Usabilidad: ¿Qué porcentaje de las acciones pueden ser realizadas automáticamente sin intervención humana?

Disponibilidad: ¿Qué porcentaje de los procesos automatizados están operativos y disponibles?

Confiabilidad: ¿Qué porcentaje de las veces los procesos automatizados operan sin fallos?

Satisfacción: ¿Qué porcentaje de los usuarios está satisfecho con el nivel de automatización disponible actualmente?

4. Obtener información del audio:

Productividad: ¿En qué porcentaje de las veces se obtiene la información correcta del audio grabado?

Usabilidad: ¿Qué porcentaje de los usuarios considera fácil obtener información del audio grabado, en una escala del 0% al 100%?

Disponibilidad: ¿Qué porcentaje de las veces el sistema está disponible para obtener la información del audio grabado?

Confiabilidad: ¿Qué porcentaje de las veces se obtiene información precisa del audio sin errores?

Satisfacción: ¿Qué porcentaje de los usuarios está satisfecho con la obtención de información del audio grabado?

Anexo 2. Cuestionario de Usabilidad basado en la ISO 25010.

Este formulario está diseñado para recoger datos sobre la experiencia del usuario sobre el Software AutoRecorder.

Cuestionario de Usabilidad basado en la ISO 25010

Este formulario está diseñado para recoger datos sobre la experiencia del usuario sobre el Software AutoRecorder

* Obligatorio

1. Nombre y Apellido *

Escriba su respuesta

2. Edad *


- Entre 15 - 17 años
- Entre 18 - 24 años
- Entre 25 - 34 años
- Entre 35 - 44 años
- Entre 45 - 54 años
- Entre 55 - 65 años

3. Sexo: * 


- Hombre
- Mujer

4. Ocupación en la radio: * 

- Administrativo
- Técnico/Operador
- Locutor

5. ¿Qué nivel de conocimiento tiene en informática? * 

- Ninguna
- Bajo
- Básico
- Avanzado

6. Comprensión de la interfaz del Software AutoRecorder. * 

| | Muy difícil | Difícil | Neutral | Fácil | Muy fácil |
|---|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| ¿Qué tan fácil le resultó comprender la interfaz del Software AutoRecorder? | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

7. Uso de las funciones principales del Software AutoRecorder. * 

| | Muy lento | Lento | Neutral | Rápido | Muy rápido |
|--|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| ¿Qué tan rápido pudo aprender a usar las funciones principales de la aplicación? | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

8. Navegación dentro Software AutoRecorder.

*

| | Muy confusa | Confusa | Neutral | Intuitiva | Muy intuitiva |
|--|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| ¿Qué tan intuitiva le pareció la navegación dentro de la aplicación? | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

9. Eficiencia de la funcionalidad de realizar respaldos de audio? *

| | Muy ineficiente | Ineficiente | Neutral | Eficiente | Muy eficiente |
|---|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| ¿Qué tan eficiente encontró la funcionalidad de realizar copias de seguridad? | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

10. Búsqueda de información en el audio de respaldo.

*

| | Muy difícil de usar | Difícil de usar | Neutral | Fácil de usar | Muy fácil de usar |
|--|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| ¿Qué tan claro y accesible encontró la función de búsqueda de información en el audio? | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

11. Tiempo para completar tareas.

*

| | Muy insatisfecho | Insatisfecho | Neutral | Satisfecho | Muy satisfecho |
|---|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| ¿Qué tan satisfecho está con el tiempo que toma la aplicación para completar tareas (Configurar la automatización, buscar información)? | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |


12. Errores o dificultades técnicas mientras usaba el Software AutoRecorder. * 

| | Frecuentemente | Algunas veces | Raramente | Nunca |
|---|-----------------------|-----------------------|-----------------------|-----------------------|
| ¿Encontró errores o dificultades técnicas mientras usaba la aplicación? | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |


13. Estética y diseño visual del Software AutoRecorder.

* 

| | Muy malo | Malo | Neutral | Bueno | Muy bueno |
|---|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| ¿Cómo calificaría la estética y diseño visual de la aplicación? | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

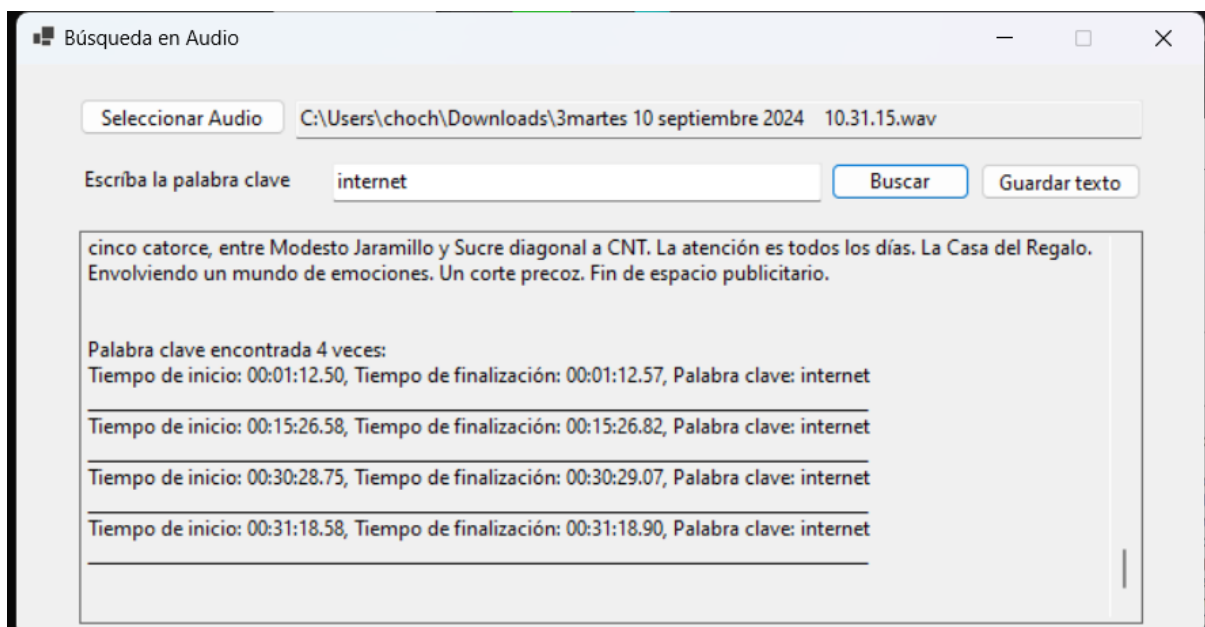
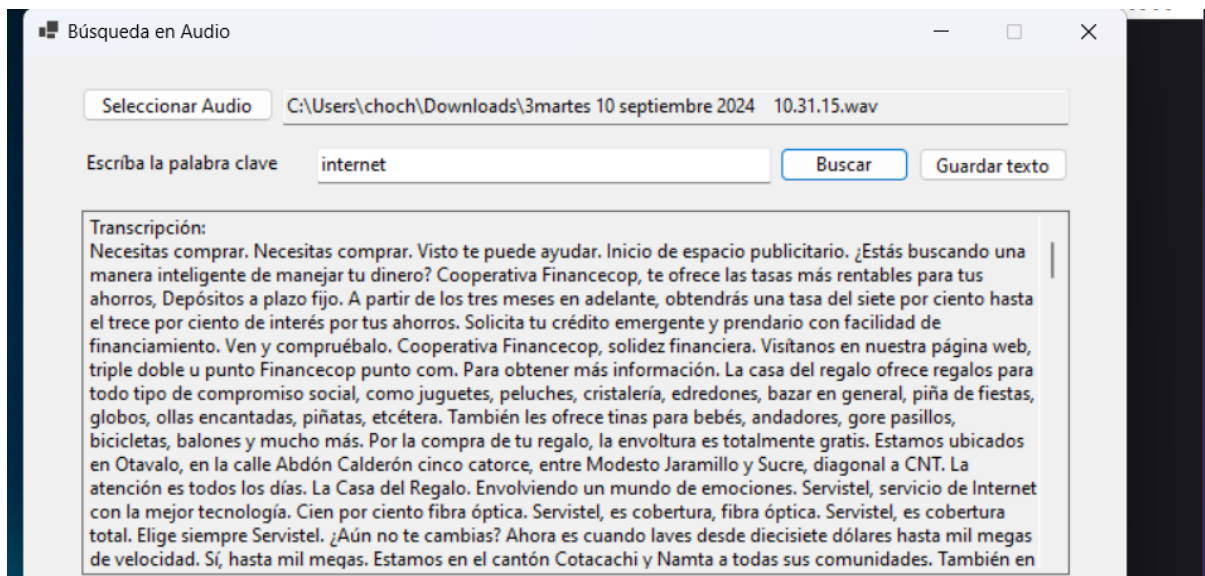
14. Resiliencia a fallos * 

| | Muy mal | Mal | Neutral | Bien | Muy bien |
|--|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| ¿Qué tan bien cree que la aplicación protege contra errores del usuario? | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

15. Recomendación a otros usuarios. * 

| | Definitivamente no | Probablemente no | No estoy seguro | Probablemente sí | Definitivamente sí |
|--|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| ¿Recomendaría esta aplicación a otras estaciones de radio? | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

Anexo 3: Software de grabación automática en ejecución.



Anexo 4: Certificado de entrega del software final.



Otavaló, 12 de septiembre del 2024

CERTIFICADO DE ENTREGA Y RECEPCION

Lic. Rodrigo Perugachi Rodríguez, director del medio de comunicación RS 105.5 FM., a petición verbal del interesado.

CERTIFICA:

Que el señor Daniel Mesías Chocho Guandinango con número de cédula 1004948871, se realizó la entrega del Software de grabación automática "AutoRecorder" del trabajo de titulación denominado: Desarrollo de una aplicación de escritorio que automatice el respaldo de grabaciones de la programación radial utilizando C# y búsqueda de información en el audio mediante técnicas de inteligencia artificial., aplicando la Norma ISO/IEC 25010, para mejorar y fortalecer la calidad de respaldos de audio.

Es todo cuanto puedo certificar en honor a la verdad, pudiendo el interesado hacer uso del presente como lo estime conveniente.

Atentamente



RODRIGO PERUGACHI
RODRIGUEZ



Lic. Rodrigo Perugachi Rodríguez
DIRECTOR DE RS 105.5 FM



DIRECCIÓN: La Compañía-edif... Tupac Amaru / Calle: Rumiñahui y Republica esquina
CONTACTOS: 0962644699 [movistar](#) / 0994890040 [claro](#)
EMAIL: runastereo@gmail.com
FACEBOOK: RS 105 5 FM - **PAGINA WEB:** www.runastereo.fm

