



UNIVERSIDAD SANTO TOMÁS
PRIMER CLAUSTRO UNIVERSITARIO DE COLOMBIA
T U N J A

TÍTULO PASANTÍA

Sistema de reconocimiento de voz basado en inteligencia artificial para la empresa Luxity
Lab

PROPONENTE(S)

Leidy Dahianna Cabrera Amortegui
CC. 1053588145
Cod. 2280553

DIRECTOR

Ing. Martha Susana Contreras Ortiz

Tunja
13 de junio del 2023

Contenido

1. FICHA TÉCNICA DE LA PASANTÍA.....	4
2. PLANTEAMIENTO DEL PROBLEMA	5
3. OBJETIVOS.....	6
3.1. OBJETIVO GENERAL	6
3.2. OBJETIVOS ESPECÍFICOS.....	6
4. DESCRIPCIÓN DE LAS ACTIVIDADES REALIZADAS.....	7
4.1. ACTIVIDADES REALIZADAS EN EL PROYECTO	7
4.1.1. FASE 1: ESTUDIO DE LAS HERRAMIENTAS A GESTIONAR	7
4.1.2. FASE 2: ANÁLISIS DE HISTORIAS DE USUARIO.....	7
4.1.3. FASE 3: IMPLEMENTAR ARQUITECTURA POR CAPAS DE LA SOLUCIÓN	8
4.1.4. FASE 4: DESARROLLO DEL COMPONENTE BACKEND (CAPA DE NEGOCIO).....	10
4.1.5. FASE 5: INTEGRAR COMPONENTES BACKEND Y FRONTEND.....	14
4.1.6. FASE 6: TESTEO DEL DESARROLLO	15
4.1.7. FASE 7: IMPLEMENTACIÓN DEL APLICATIVO PARA PRODUCCIÓN	16
4.1.8. FASE 8: DESPLIEGUE DEL DESARROLLO	16
4.2. HERRAMIENTAS UTILIZADAS EN EL DESARROLLO DEL PROYECTO	17
5. CONCLUSIONES	18
6. ANEXOS.....	20
CERTIFICADO DE FINALIZACIÓN	20
ANEXO 1. MODELADO CONCEPTUAL DEL NEGOCIO	21
ANEXO 2. DOCUMENTO DE ESPECIFICACIÓN DE REQUERIMIENTOS - CAPA LÓGICA.....	24
ANEXO 3. DOCUMENTACIÓN TÉCNICA	36

ANEXO 4. INFORME DE TESTEO.....	39
7. REFERENCIAS.....	42

1. FICHA TÉCNICA DE LA PASANTÍA

Título	Sistema de reconocimiento de voz basado en inteligencia artificial para la empresa Luxity Lab.
Nombre Estudiante	Leidy Dahianna Cabrera Amortegui
Documento estudiante	1053588145
Correo electrónico estudiante	Leidy.cabrera@usantoto.edu.co
Director	Martha Susana Contreras Ortiz
Entidad o sector donde realizará la pasantía	Luxity Lab
Lugar de ejecución de la pasantía	Tunja Boyacá (Remoto)
Duración	6 meses
Los abajo firmantes confirman que todos los datos incluidos en la presente propuesta son correctos y verídicos, que no incumplen ninguna ley o norma vigente (incluir nombres y firmas de estudiantes y director).	
 Firma del autor Leidy Dahianna Cabrera Amortegui  Firma del director Ing. Martha Susana Contreras Ortiz	

2. PLANTEAMIENTO DEL PROBLEMA

En la actualidad, los sistemas están basados en distintos tipos de mecanismos de autenticación, por ejemplo: los que están basados en usuario/contraseña, en tokens, en certificados digitales o mecanismos biométricos (Sepúlveda Marín, 2022). Sin embargo, los sistemas informáticos se han visto afectados en gran medida por diferentes tipos de deficiencias y ataques. De los mecanismos de autenticación mencionados, los más afectados por los ataques cibernéticos son los basados en usuario/contraseña dentro de formularios.

Según la investigación realizada en (Internet Security Auditors, 2005), las deficiencias y ataques que sufren este tipo de mecanismos están centrados en los siguientes problemas: fugas de información, debilidad en los campos de los formularios de autenticación, deficiencia de las funcionalidades como en el registro de usuarios, modificación de los datos o recuperación de contraseñas. Una de las deficiencias más frecuentes es en el sistema de gestión de sesiones, teniendo en cuenta el tiempo de vida limitado para el token y la utilización de algoritmos débiles en la generación de los mismos. Las consecuencias que traen consigo las problemáticas mencionadas anteriormente posibilitan en gran medida el sniffing (ciberataque que analiza, monitoriza, y captura datos), phishing (engaño a usuarios con el fin de que compartan información como contraseñas o números de tarjetas de crédito), suplantación de personalidad u obtención de información sensible.

3. OBJETIVOS

3.1. Objetivo General

Desarrollar el backend de un sistema de reconocimiento de voz basado en Inteligencia Artificial de Phonexia Voice Verify para la autenticación por voz en plataformas web y móviles.

3.2. Objetivos específicos

Tabla 1. Objetivos específicos

Nro.	Objetivo específico
1	Estructurar el modelo conceptual del negocio a partir de las historias de usuario definidas para el sistema de autenticación biométrica de la empresa Luxity Lab.
2	Construir la capa lógica del negocio de la arquitectura basada en capas para dar cumplimiento al modelo conceptual, utilizando el lenguaje de programación Python.
3	Implementar la solución informática aplicando la arquitectura basada en capas e integrando los servicios de Voice Verify de Phonexia, Twilio, AWS para el procesamiento de autenticación por voz.
4	Aplicar un plan de pruebas definidas por la empresa para medir la calidad del desarrollo.

Fuente: Autor

4. DESCRIPCIÓN DE LAS ACTIVIDADES REALIZADAS

4.1. Actividades realizadas en el proyecto

Las actividades realizadas en el transcurso de este proyecto, fueron divididas por fases (1-8) y serán presentadas de la misma forma.

4.1.1. Fase 1: estudio de las herramientas a gestionar

Las actividades pertenecientes en esta fase se llevaron a cabo durante las tres primeras semanas del mes de noviembre del año 2022. En estas semanas se realizó una capacitación en torno a las tecnologías Twilio (Twilio, 2023) y Phonexia (Phonexia, 2023), como requisito de la empresa, con el fin de tener los conocimientos suficientes para el desarrollo del proyecto. El método utilizado para realizar dichos estudios fue de manera individual y por capacitaciones dadas en reuniones por la plataforma Google Meet por parte de la empresa. A continuación, una tabla con el detalle de las actividades:

Tabla 2. Actividades fase 1

Descripción de la actividad	Fecha inicio	Fecha entrega	Producto entregado
Desarrollo curso Django	01/11/2022	04/11/2022	Certificado de finalización
Estudio de la librería OAut2	07/11/2022	11/11/2022	Información para realización del modelado conceptual del negocio
Estudio de tecnologías Twilio, Phonexia, AWS	14/11/2022	18/11/2022	
Análisis de recursos necesarios para despliegues en las diferentes tecnologías			

Fuente: Autor

4.1.2. Fase 2: análisis de historias de usuario

Esta fase cuenta con una actividad y fue llevada a cabo durante la cuarta semana del mes de noviembre del año 2022. En esta semana se realizaron reuniones diarias por medio de la plataforma Google Meet con el jefe de proyecto y desarrollador frontend. Por medio de las mismas, se lograron establecer los requerimientos para la realización del modelo conceptual del negocio. Para más detalles, visualizar Anexo 1. Modelo conceptual del negocio y Anexo 2. Documento de especificación de requerimientos - capa lógica.

Tabla 3. Actividades fase 2

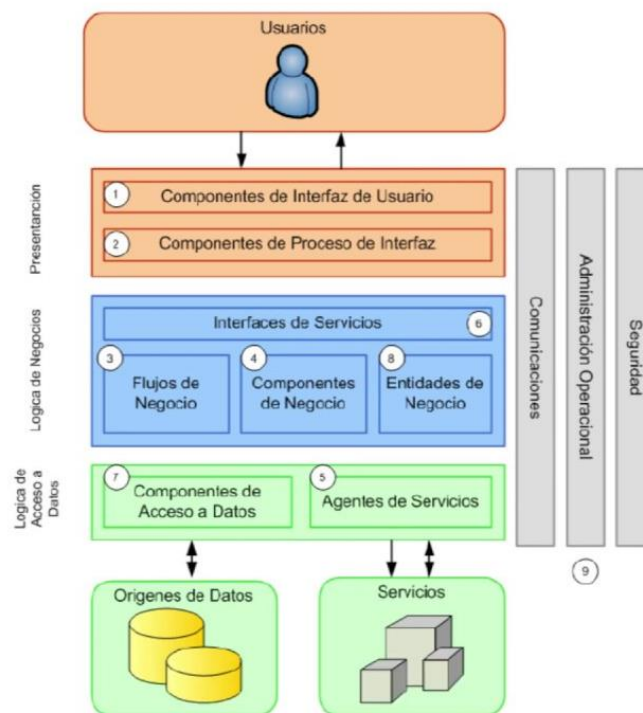
Descripción de la actividad	Fecha inicio	Fecha entrega	Producto entregado
Analizar la especificación de historias de usuario	21/11/2022	25/11/2022	Modelado conceptual del negocio, documento de especificación de requerimientos capa lógica

Fuente: Autor

4.1.3. Fase 3: implementar arquitectura por capas de la solución

Dando una pequeña introducción, la arquitectura basada en capas es un enfoque de diseño que divide un sistema en capas funcionales (capa de presentación, capa lógica, capa de persistencia) que pueden ser desarrolladas y probadas de forma independiente, lo que permite una estructura modular, una gestión más sencilla y la posibilidad de reutilizar componentes. A continuación, una imagen que nos permite comprender un poco más el detalle acerca de la misma:

Figura 1. Arquitectura por capas



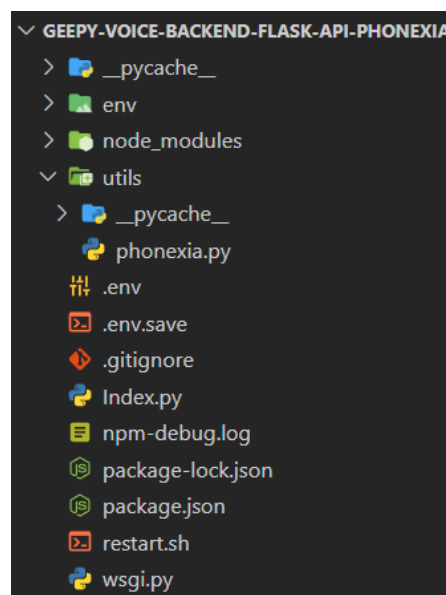
Fuente: (Likecomtic - Arquitectos de Software, 2020)

Las actividades realizadas dentro de esta fase corresponden a la planeación que fue establecida por medio de reuniones mediante la plataforma Google Meet, con el objetivo de establecer qué tipo de arquitectura implementar. Estas actividades fueron desarrolladas en la cuarta semana del mes de noviembre del año 2022. En conclusión, se estableció el modelo arquitectónico por capas, teniendo en cuenta que varias áreas dentro de la empresa estarían interviniendo en el proyecto. En el caso particular de esta pasantía, se asignó la responsabilidad del desarrollo de la capa lógica del negocio. Además, se asignó una parte de la capa de persistencia, teniendo en cuenta que es necesario el uso de la misma para el desarrollo de la capa de negocio.

Como producto de las reuniones, se llegó a la conclusión que la capa lógica del negocio tendría la siguiente estructura (Ver Figura 2):

- **env**: entorno de Python con intérprete.
- **node_modules**: carpeta que almacena librerías complementarias en el proyecto
- **utils**: administra el componente Phonexia con todos los respectivos métodos para hacer uso de las Inteligencias Artificiales (IA).
- **.env**: almacena variables de entorno.
- **Index.py**: archivo Python con métodos principales de ejecución.

Figura 2. Componentes capa lógica



Fuente: Autor

Tabla 4. Actividades fase 3

Descripción de la actividad	Fecha inicio	Fecha entrega	Producto entregado
Analizar la arquitectura de la solución	28/11/2022	02/12/2022	Creación proyecto Python con capa lógica
Revisar documentación de la arquitectura			

Fuente: Autor

4.1.4. Fase 4: desarrollo del componente backend (capa de negocio)

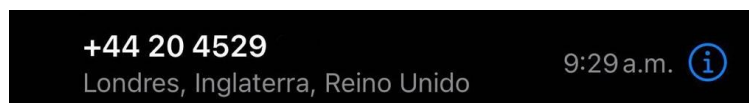
La capa de negocio fue desarrollada en el transcurso de las semanas 5 a 18. Las actividades realizadas dentro de esta fase permitieron obtener toda la capa lógica.

Al inicio de esta fase, se estructuró la base datos. Esta consta de 5 tablas (Voice signature, connection id, Voice oauth2, user, text to speak) las cuales podrán ser visualizadas como modelo relacional dentro del Anexo 3. Documentación técnica. El diagrama modelo relacional fue diseñado con la aplicación MySQL Workbench 8.0 CE (Oracle, 2023). La base de datos almacenará datos de control, debido a que Phonexia proporcionó otra base de datos que almacena las huellas de voz. A continuación, se describe en detalle cada tabla mencionada de la base de datos:

- **Voice signature:** almacena la información suministrada por cada usuario para su huella de voz, por ejemplo: número de teléfono y código de área.
- **Connection id:** almacena las conexiones temporales generadas por AWS (Amazon Web Services), que permiten la correcta transmisión del flujo de datos entre la capa de presentación y la capa de negocio.
- **Voice oauth2:** almacena el historial del proceso de cada usuario, por ejemplo, el estado en el que se encuentra la llamada (registro/enrollamiento o verificación) y la cantidad de paquetes recibidos (indicando segundo a segundo el estado del proceso).
- **User:** almacena toda la información relevante de cada usuario dentro de toda la aplicación de Geepy Cloud Connect.
- **Text to speak:** almacena diferentes recomendaciones de texto para mostrar en pantalla durante la llamada. Son consultadas aleatoriamente con el fin que los usuarios no permanezcan en silencio durante la llamada y de esta manera tener un mejor registro de la voz.

Una vez se obtuvo el diseño de la base de datos, se procedió a la programación de los componentes y servicios. El primer objetivo dentro de esta fase fue la realización de llamadas, las cuales fueron administradas y realizadas por la plataforma Twilio. Mediante Twilio es posible comprar un número de teléfono con indicativo de cualquier parte del mundo, con el cual se pueden hacer las llamadas. Para este ejercicio se compró un número telefónico de Londres, Inglaterra y Reino Unido. Por motivos de privacidad, se ocultan lo últimos dígitos (Figura 3).

Figura 3. Número telefónico Twilio



Fuente: Autor

Luego, se realizó el desarrollo correspondiente para transmitir el flujo de voz desde Twilio a Phonexia y abrir un stream. Para transmitir este flujo de datos, se inicia el stream con Twilio por medio de un websocket (IBM, 2023) que se define como una sesión de comunicación interactiva entre usuario y servidor, para luego empezar a recibir paquetes de Twilio que se pueden diferenciar por eventos (start, stop, media). Cuando el evento es start, inicia de igual forma el proceso con Phonexia. Comienza el stream y conexión con un webhook que según (Red Hat, 2023) se puede definir como una función tipo HTTP para que la API de Twilio y la nuestra se comuniquen más fácil. Se verifica qué tipo de stream es (enroll/verify). Seguido de lo anterior, cuando el usuario comienza a hablar, el evento cambia a media, y allí se envían los paquetes generados por Twilio directamente a Phonexia. Inmediatamente Phonexia empieza a responder con el estado en segundos por proceso, dependiendo el tipo (enroll/verify).

Si el evento es enroll nos devolverá segundo a segundo el estado. Teniendo en cuenta que Phonexia exige un mínimo de tiempo (15 segundos) para registrar correctamente la firma de voz, se podrán observar los paquetes teniendo en cuenta que son segundo a segundo de la siguiente manera:

Figura 4. Paquete por segundo de tipo enroll

```

>{"channel_uuid": "4afd8bd4-8712-46d9-94cf-a664c53d414e", "type":
'enroll", "payload": {"stream_uuid": "4afd8bd4-8712-46d9-94cf-
a664c53d414e", "external_id": "24", "detail": "Stream contains 14s of
minimum 15s of speech.", "speech_length": 14.0, "source_speech_quality":
null, "source_speech_quality_verdict": null}, "status": 411}'

```

Fuente: Autor

Una vez se completan los 15 segundos requeridos, Phonexia responde con el siguiente paquete:

Figura 5. Paquete de creación del voiceprint

```

>{"channel_uuid": "4afd8bd4-8712-46d9-94cf-a664c53d414e", "type":
'enroll", "payload": {"stream_uuid": "4afd8bd4-8712-46d9-94cf-
a664c53d414e", "external_id": "24", "detail": "Voiceprint created and added
to the database.", "speech_length": 15.0, "source_speech_quality": null,
"source_speech_quality_verdict": null}, "status": 201}'

```

Fuente: Autor

Para el caso de la verificación, solo se necesitan 3 segundos para que Phonexia arroje un resultado y de igual forma, llega el paquete segundo a segundo de la siguiente manera:

Figura 6. Paquete por segundo de tipo verify

```

>{"channel_uuid": "4afd8bd4-8712-46d9-94cf-a664c53d414e", "type":
{'channel_uuid': '996257e2-c009-4848-8396-627a513ef454', 'type': 'verify',
'payload': {'stream_uuid': '996257e2-c009-4848-8396-627a513ef454',
'external_id': 'd-120', 'detail': 'Stream contains 1s of minimum 3s of
speech.', 'speech_length': 1.0, 'source_speech_quality': None,
'source_speech_quality_verdict': None}, 'status': 411}'

```

Fuente: Autor

Cuando se logra la verificación Phonexia responde con el siguiente paquete:

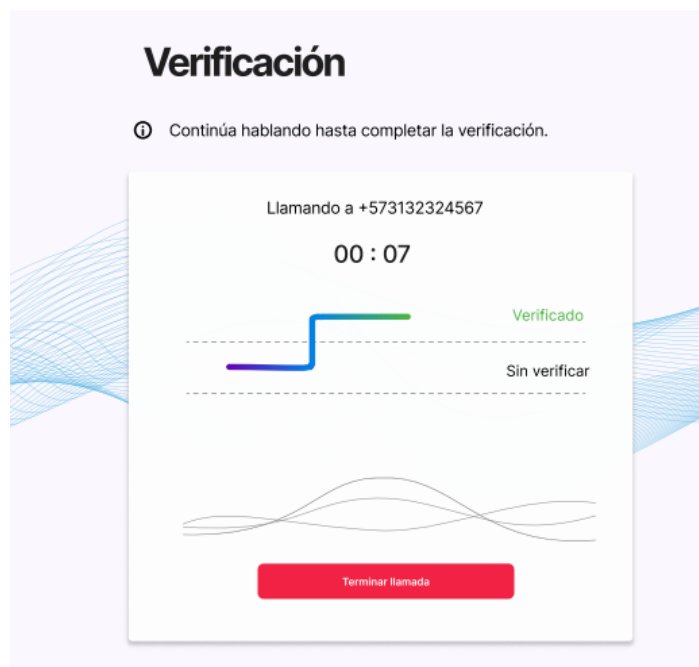
Figura 7. Paquete con respuesta de verificado

```
{'channel_uuid': '996257e2-c009-4848-8396-627a513ef454', 'type': 'verify',  
  'status': 200, 'payload': {'stream_uuid': '996257e2-c009-4848-8396-  
627a513ef454', 'external_id': 'd-120', 'result': 'verified',  
  'speech_length': 3.0, 'score': 9.9999, 'source_speech_quality_voiceprint':  
  None, 'source_speech_quality_verdict_voiceprint': None,  
  'source_speech_quality_stream': None,  
  'source_speech_quality_verdict_stream': None}}
```

Fuente: Autor

En la (Figura 8) se puede observar el resultado (verificado, no verificado) con un score de 0 a 10.0. Estos resultados son analizados por el frontend y de acuerdo a ellos son graficados en pantalla de la siguiente forma:

Figura 8. Pantalla de verificación



Fuente: Autor

En la Tabla 5 se resumen las actividades realizadas durante la Fase 4, las fechas en las cuales se desarrollaron y el producto resultante.

Tabla 5. Actividades fase 4

Descripción de la actividad	Fecha inicio	Fecha entrega	Producto entregado
Realizar análisis de historias de usuario para posterior diseño de base de datos	05/12/2022	09/12/2022	Programación de la capa lógica
Diseño del modelo de base de datos			
Desarrollo de componentes y servicios	12/12/2022	16/12/2022	
Desarrollo de integración con la tecnología Twilio para realizar llamadas	12/12/2022	06/01/2023	
Establecer conexión con la API suministrada por Phonexia	09/01/2023	03/02/2023	
Realizar pruebas de llamadas para verificación de stream de voz			
Integrar de las inteligencias artificiales Twilio y Voice Verify de Phonexia para transferir stream de voz	30/01/2023	24/02/2023	
Realizar inscripción y verificación de la huella de voz en Phonexia	06/02/2023	03/03/2023	
Análisis e interpretación de resultados arrojados por Phonexia	27/02/2023	10/03/2023	

Fuente: Autor

4.1.5. Fase 5: integrar componentes backend y frontend

Esta fase fue desarrollada en el transcurso de la semana 19. La integración de ambas partes fue desarrollada en conjunto con el área encargada del frontend y la gerencia de la empresa. El despliegue se realizó por medio de Amazon Web Services y el objetivo principal fue empezar a hacer pruebas de conexión con las dos APIs, verificando que los paquetes de respuesta de Phonexia llegaran en tiempo real. Una vez logrado ésto, se realizaron reuniones diarias con el equipo de desarrollo, con el fin de establecer un plan de pruebas entre ambas partes para corrección de posibles fallas y componentes faltantes.

Tabla 6. Actividades fase 5

Descripción de la actividad	Fecha inicio	Fecha entrega	Producto entregado
Despliegue de solución en AWS	13/03/2023	17/03/2023	API en producción
Integrar frontend y backend			

Fuente: Autor

4.1.6. Fase 6: testeo del desarrollo

Esta fase fue desarrollada en las semanas 20 y 21. La empresa brindó al equipo de desarrollo las orientaciones sobre cuáles pruebas realizar. A medida que el equipo de desarrollo realizaba las validaciones incluidas en el plan de pruebas, ellos debían diligenciar todas las incidencias detectadas en un instrumento de Excel compartido. Para ello, se registraban los siguientes datos:

- **Consecutivo:** Número de la incidencia
- **Módulo:** inicio de sesión, principal, registro o verificación
- **Componente:** especificación de la ubicación del error, por ejemplo, creación de huella de voz.
- **Descripción:** describe de una forma corta y concreta el error
- **Solucionado:** se registra “ok” o “en curso”

Para más detalles de los incidentes encontrados, visualizar Anexo 4. Informe de testeo.

En la siguiente tabla se podrán evidenciar las actividades realizadas en el transcurso de la fase 6, con sus respectivas fechas y productos entregables:

Tabla 7. Actividades fase 6

Descripción de la actividad	Fecha inicio	Fecha entrega	Producto entregado
Analizar el programa de testeo suministrado por la empresa	20/03/2023	31/03/2023	Informe de testeo
Inicio de testeo de la API			
Testeo por segunda vez para verificar correcciones			
Testeo final			

Fuente: Autor

4.1.7. Fase 7: implementación del aplicativo para producción

Esta fase fue llevada a cabo en el transcurso de la semana 22. Una vez identificadas las incidencias, se realizaron los respectivos cambios tanto en el área de backend como de frontend, con el objetivo de realizar un testeo por segunda vez de la aplicación, obteniendo como resultado la finalización exitosa del plan de pruebas.

En esta fase, se han incorporado nuevos servicios a Geepy Voice con el objetivo de ofrecer una forma fácil y efectiva de autenticación por voz. Estos servicios adicionales se han diseñado específicamente para iniciar sesión a través de la autenticación por voz y para su posterior implementación en cualquier otra plataforma. Para demostrar la utilidad del mismo, se implementaron estos servicios en la plataforma de Geepy Cloud Connect reemplazando la forma tradicional de inicio de sesión (usuario/contraseña) por el sistema de autenticación por voz (Geepy Voice) con esta implementación, se pudo concluir lo siguiente: el desarrollo del sistema de autenticación proporciona mayor seguridad y rapidez en el inicio de sesión, teniendo en cuenta que solo se requieren 3 segundos de llamada para permitir el acceso a la aplicación.

Tabla 8. Actividades fase 7

Descripción de la actividad	Fecha inicio	Fecha entrega	Producto entregado
Realizar correcciones halladas en el testeo	03/04/2023	07/04/2023	Programación capa lógica
Desarrollar componentes adicionales para una versión de venta			

Fuente: Autor

4.1.8. Fase 8: despliegue del desarrollo

Esta fase se llevó a cabo durante las semanas 23 y 24. Para ello se realizó el acompañamiento en el despliegue, teniendo en cuenta que dichos procesos son realizados directamente por la gerencia de la empresa.

A continuación, se puede observar la Tabla 9 con la descripción de las actividades que corresponden a la fase 8 de la pasantía, en las que se puede encontrar todas las actividades realizadas con fechas y productos entregados.

Tabla 9. Actividades fase 8

Descripción de la actividad	Fecha inicio	Fecha entrega	Producto entregado
Configurar ambiente de producción	10/04/2023	21/04/2023	Demo Geepy Voice funcional.
Realizar despliegue en AWS con el ambiente configurado			

Fuente: Autor

4.2. Herramientas utilizadas en el desarrollo del proyecto

Tabla 10. Herramientas utilizadas

Herramienta utilizada	Para qué la utilizó
Visual Studio Code	Editor de código fuente del proyecto
PostgreSQL	Sistema gestor de base de datos, donde se tiene la base de datos principal de Geepy Cloud Connect.
Ngrok	Permitió crear un túnel seguro entre el servidor local y Twilio para la recepción de paquetes.
Python	Lenguaje de programación interpretado. Fue el lenguaje principal para el desarrollo del proyecto.
Flask	Es un microframework web para Python que facilitó la creación de la aplicación web de manera rápida y sencilla.
AWS	Plataforma de servicios en la nube con infraestructura estable para realizar el despliegue del proyecto.
Twilio	Plataforma de comunicaciones en la nube que permitió integrar y gestionar las llamadas de autenticación.
Phonexia Voice Verify	IA de verificación de voz basada en tecnología de reconocimiento de voz. Permite autenticar y verificar la voz de una persona utilizando la voz como factor de autenticación.
Postman	Plataforma para probar documentar y realizar solicitudes HTTP de manera fácil y eficiente.

5. CONCLUSIONES

Para concluir, el desarrollo del backend o capa de negocio del proyecto Geepy Voice para la empresa Luxity Lab culminó satisfactoriamente, gracias a la implementación de tecnologías de reconocimiento de voz como lo es Phonexia Voice Verify, Twilio como plataforma de desarrollo de las comunicaciones y a un correcto ciclo de desarrollo de software.

Al analizar y comprender las historias de usuario, se lograron identificar los requisitos clave, las funcionalidades y las interacciones necesarias para desarrollar el sistema de autenticación, incluyendo a este, el proceso de estructurar el modelo conceptual del negocio que implicó la creación de una representación visual o diagrama en el que se puede visualizar cómo se interrelacionan los diferentes componentes del sistema, como los usuarios, las tecnologías de Twilio (Twilio, 2023), Phonexia (Phonexia, 2023), la base de datos y los procesos de autenticación. Esto proporcionó una visión clara de cómo se debían diseñar, desarrollar e integrar estos componentes para lograr una solución completa y coherente.

Durante la construcción de la capa lógica del negocio, se presentaron grandes desafíos técnicos como lograr la transmisión del flujo de datos en tiempo real a la API de Phonexia para su posterior tratamiento, teniendo en cuenta que Twilio posee poca documentación. Pese a dicha problemática, se han logrado aplicar diversas técnicas de solución de problemas facilitando el procesamiento de voz para extraer y comparar características clave de la voz de los usuarios.

La arquitectura basada en capas fue de gran utilidad, teniendo en cuenta que en el desarrollo del proyecto estuvieron involucradas diferentes áreas de la empresa. Esta arquitectura brindó modularidad, separación de preocupaciones, escalabilidad, facilidad de mantenimiento, abstracción y flexibilidad. Estos beneficios contribuyeron a un desarrollo más eficiente, una mayor legibilidad del código y una adaptabilidad del sistema a medida que evoluciona con el tiempo.

Al utilizar la voz como factor de autenticación, se ha brindado a los usuarios una forma más natural y fácil de acceder a sus cuentas y proteger su información personal. Además, se han implementado medidas de seguridad adicionales como el doble factor de autenticación con un número telefónico de confianza para garantizar la integridad y confidencialidad de los datos de los usuarios.

La aplicación del plan de pruebas definido por la empresa fue esencial para garantizar que el sistema cumple con los estándares y requisitos establecidos. Mediante la realización de pruebas exhaustivas y sistemáticas, se pudieron identificar y corregir errores o deficiencias en el sistema, lo que contribuyó a mejoras en calidad y fiabilidad antes de su implementación final.

En general, cada una de las fases que comprendió el desarrollo de Geepy Voice, fueron de vital importancia en el desarrollo del mismo, permitiendo así culminar el desarrollo en el tiempo establecido.

6. ANEXOS

Certificado de finalización



Número de certificado: UC-b60256c0-2b04-4376-bc67-1af400bb70b8
Url del certificado [ude.my/UC-b60256c0-2b04-4376-bc67-1af400bb70b8](https://udemy.com/certificate/UC-b60256c0-2b04-4376-bc67-1af400bb70b8)
Número de referencia: 0004

CERTIFICADO DE FINALIZACIÓN

Curso Django y Django rest framework de básico a profesional

Instructores **Cristhian Santa Cruz**

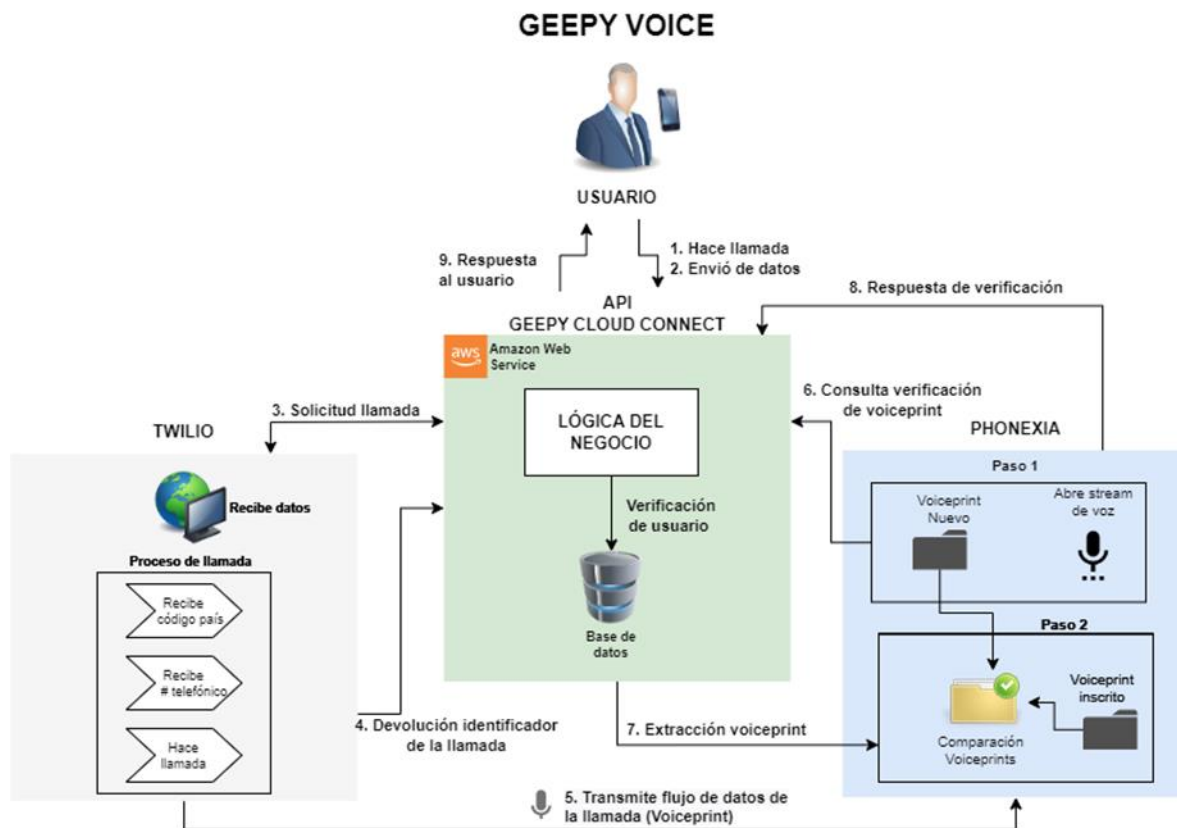
Dahianna Cabrera

Anexo 1. Modelado conceptual del negocio

MODELADO CONCEPTUAL DEL NEGOCIO

Geepy Voice

Figura 1. Modelo conceptual



Fuente: Autor

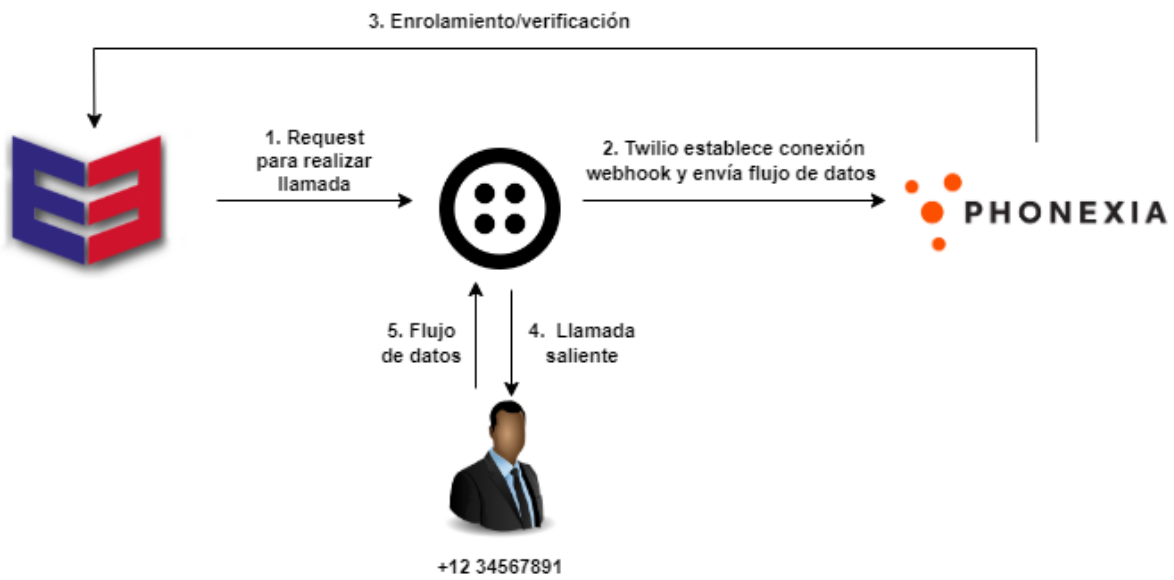
Funcionalidad de cada componente

Twilio: Plataforma con API flexible que permite realizar las llamadas y redireccionar el flujo de datos a Phonexia Voice Verify.

Phonexia Voice Verify: Solución basada en biometría de voz para la autenticación y verificación de usuarios, su API nos permite la inscripción, enrolamiento/verificación de voiceprints (Huellas de voz).

API Geepy Cloud Connect: Su funcionalidad principal con Twilio es administrar y disparar las llamadas de tal forma que su flujo de datos sea redireccionado a Phonexia. Con Phonexia, su fin es abrir streams, enviar, enrolar/verificar voiceprints (Huellas de voz), cerrar streams. Por último, su base de datos propia permite la administración de las inscripciones de voz por usuario.

Figura 2. Modelo conceptual parcial



Fuente: Autor

Anexo 2. Documento de especificación de requerimientos - capa lógica

DOCUMENTO DE ESPECIFICACIÓN DE REQUERIMIENTOS - CAPA LÓGICA

Geepy Voice

Hoja de control del documento

Título	Documento de especificación de requerimientos	
Responsable	Camilo Cano	
Aprobado por	Camilo Cano	Fecha aprobación 25/11/2022
Versión	1.0	Fecha versión 25/11/2022

Participantes de elaboración

Elaboradores	Unidad Organizativa
Leidy Dahianna Cabrera Amortegui	Universidad Santo Tomás Seccional Tunja – Facultad de Ingeniería de sistemas Estudiante
Camilo Cano	Luxity Lab Gerente

Registro de cambios

Versión	Causa del cambio	Responsable del cambio	Fecha cambio
1.0	Creación del documento	Leidy Dahianna Cabrera Amortegui	21/11/2022

CONTENIDO

1. INTRODUCCIÓN.....	27
1.1. OBJETO.....	27
1.2. ÁMBITO.....	27
2. DESCRIPCIÓN GENERAL DEL SISTEMA.....	27
2.1. DESCRIPCION DE LA SITUACIÓN ACTUAL.....	27
2.2. CARACTERÍSTICAS PRINCIPALES.....	27
3. CATÁLOGO DE REQUISITOS.....	33
4. REQUISITOS FUNCIONALES.....	34

1. INTRODUCCIÓN

1.1. OBJETO

Construir un sistema informático que facilite la autenticación por medio de la voz, permitiendo el registro y verificación de voz dando uso a las plataformas Twilio, Phonexia Voice Verify y la API de Geepy Cloud Connect.

1.2. ÁMBITO

Aplicativo para uso de la empresa Geepy Cloud Connect.

2. DESCRIPCIÓN GENERAL DEL SISTEMA

2.1. DESCRIPCIÓN DE LA SITUACIÓN ACTUAL

Por propuesta de la empresa se solicita la creación del aplicativo como una posible oportunidad de negocio.

2.2. CARACTERÍSTICAS PRINCIPALES

Ingreso a la plataforma DEMO

Pantalla de inicio de sesión del demo:



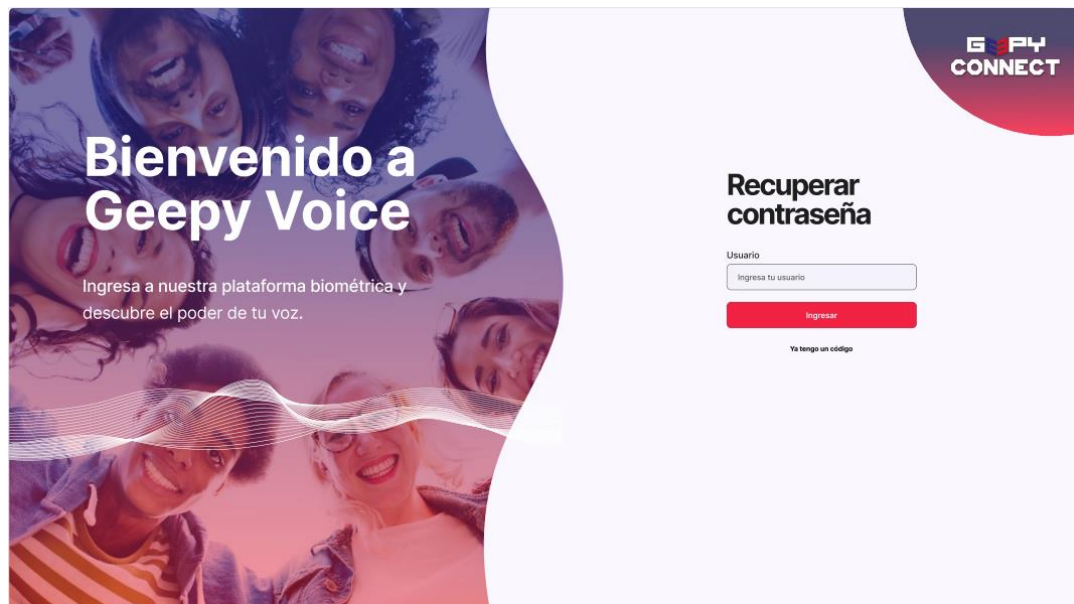
El usuario debe ingresar correo electrónico, contraseña y aceptar términos y condiciones de la empresa Geepy Cloud Connect para el ingreso al demo.

El usuario deberá llenar un formulario con los siguientes campos:

Correo electrónico: Correo registrado en Geepy Cloud Connect.

Contraseña: Contraseña registrada en Geepy Cloud Connect.

Recuperación de contraseña



El usuario a hacer prueba podrá recuperar contraseña en caso de que sus credenciales sean muy antiguas, recibirá un correo con código de restablecimiento de contraseña. En caso de recuperación es necesario ingresar el correo electrónico.

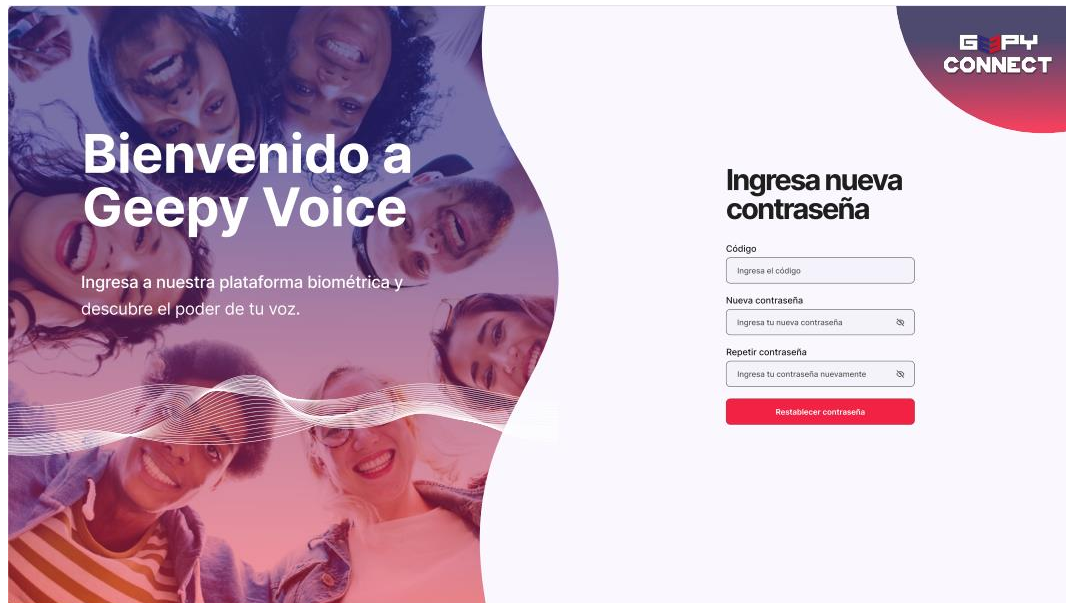
Cambio de contraseña

Una vez el usuario ingresa el código (válido) que ha sido enviado a su correo electrónico, se deberá llenar los siguientes campos en el formulario:

Código: Código enviado a correo electrónico.

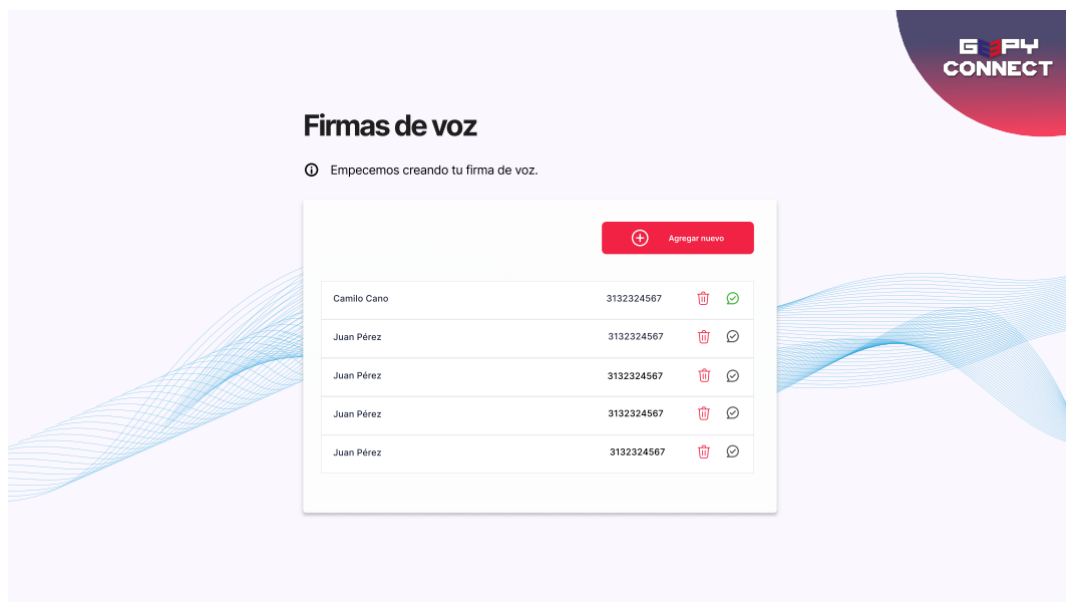
Nueva contraseña: Ingreso de nueva contraseña de al menos 8 caracteres.

Repetir nueva contraseña: Ingreso nuevamente de la nueva contraseña.



Pantalla principal

El usuario podrá encontrar todas las firmas de voz guardadas en la base de datos, diferenciándolas según el nombre y el número de teléfono registrado. De igual forma permite agregar nuevas firmas.



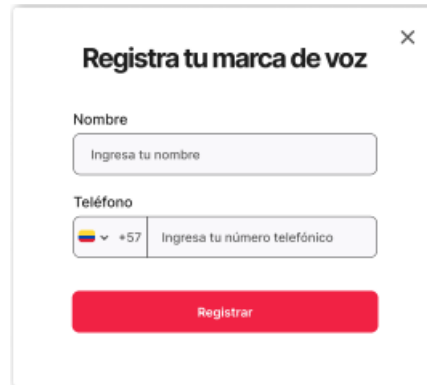
Formulario para registro de firma de voz

El usuario debe ingresar los siguientes datos en el formulario:

Nombre: Nombre para identificación de la firma de voz.

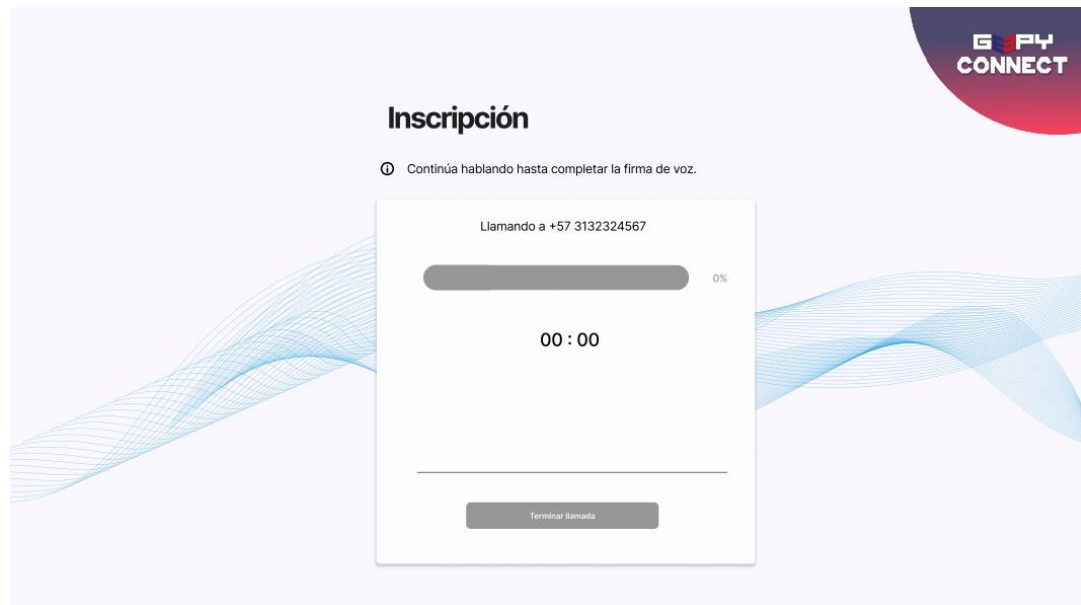
Código del país: Código del país relacionado con el número de teléfono.

Número de teléfono: Número de teléfono a llamar para realizar proceso de autenticación.

El formulario se titula "Registra tu marca de voz" y tiene un botón de cerrar en la esquina superior derecha. Contiene dos campos de entrada: "Nombre" con el placeholder "Ingresa tu nombre" y "Teléfono" que incluye un selector de bandera (España) y el código "+57" con el placeholder "Ingresa tu número telefónico". Debajo de los campos hay un botón rojo con el texto "Registrar".

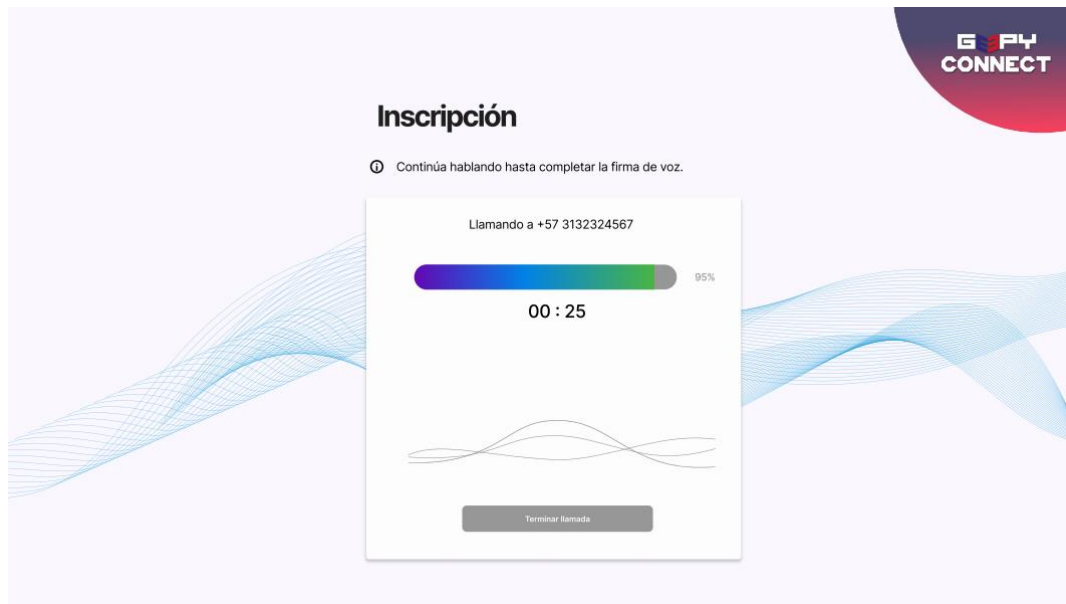
Inscripción de huella de voz

Para realizar la inscripción de huella de voz, el usuario debe responder la llamada y comenzar a hablar.

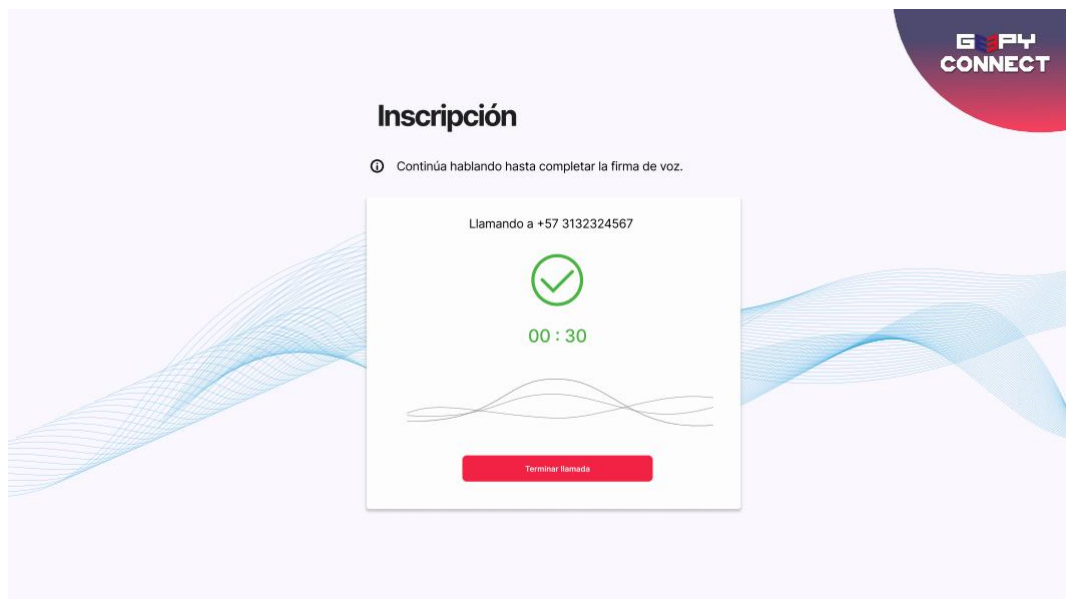
La pantalla tiene un fondo con ondas azules y el logo "G4PY CONNECT" en la esquina superior derecha. El título principal es "Inscripción". Debajo de él hay un ícono de información y el texto "Continúa hablando hasta completar la firma de voz.". En el centro hay un recuadro blanco que muestra "Llamando a +57 3132324567", una barra de progreso gris que indica "0%", un temporizador "00 : 00" y un botón gris "Terminar llamada".

En pantalla podrá observar el tiempo de conexión a la llamada. Es importante recalcar, que el tiempo mínimo para realizar la inscripción son 15 segundos y

debe permanecer hablando el mayor tiempo posible sin ruidos de fondo para obtener una huella más precisa.



Cuando la inscripción sea exitosa, el usuario podrá observar la siguiente pantalla.



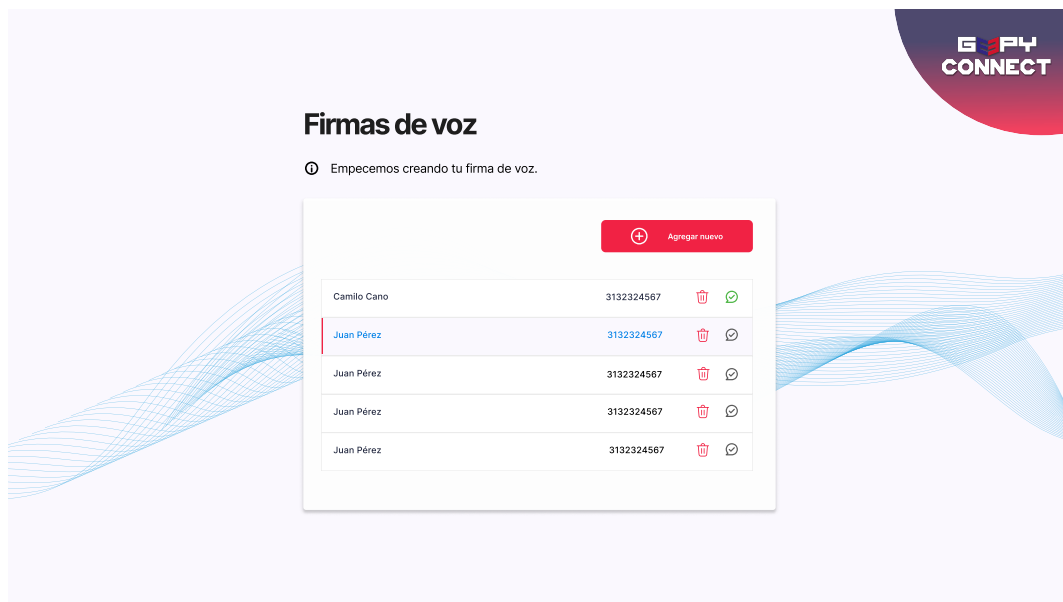
Verificación de huella de voz

En pantalla principal se podrán observar dos iconos al lado derecho del número telefónico.

 : debe permitir eliminar una firma de voz.

 : firma de voz sin verificación y redirige al formulario para iniciar llamada de verificación.

 : firma de voz con verificación completa.



Formulario de verificación

El usuario podrá observar el tamaño de la firma inscrita y visualizar datos para posteriormente iniciar llamada.



Juan Pérez +573132324567

Tamaño de la firma: 30.05s

Teléfono

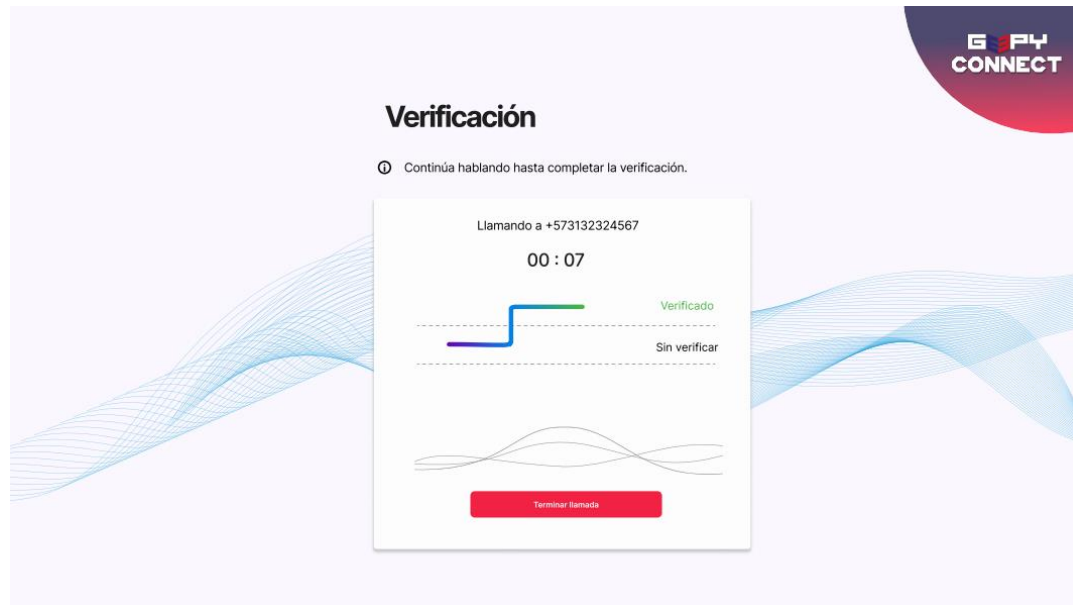
+573132324567

Prepara tu dispositivo celular para recibir la llamada.

Eliminar **Verificar**

Estado de la verificación

El usuario en llamada podrá observar el estado de su verificación. Si su voz coincide con la de la firma de voz registrada, el resultado será **verificado**, de lo contrario, la línea de estado se posicionará en **sin verificar**.



3. CATÁLOGO DE REQUISITOS

Id. Requisito	Descripción corta	Tipo (F, NF, TC)
01	Inicio de sesión al demo	F
02	Recuperación de contraseña	F
03	CRUD firmas de voz	F
04	Inscripción/enrolamiento de huella de voz	F
05	Verificación de huella de voz	F
06	CRUD texto para hablar	F

F: requisito funcional, **NF:** requisito no funcional, **TC:** requisito técnico

4. REQUISITOS FUNCIONALES

Número del requisito	01
Tipo	☒ Requisito ☐ Restricción
Fuente del requisito	
Prioridad del requisito	☒ Alta/Eencial ☐ Media/Deseado ☐ Baja/ Opcional
Descripción del requisito	Para iniciar sesión en el demo es necesario ingresar usuario, contraseña y aceptar términos y condiciones de Geepy Cloud Connect se debe llenar un formulario y validar estos campos.

Número del requisito	02
Tipo	☒ Requisito ☐ Restricción
Fuente del requisito	
Prioridad del requisito	☒ Alta/Eencial ☐ Media/Deseado ☐ Baja/ Opcional
Descripción del requisito	Para realizar cambios en la contraseña se debe solicitar un código que será enviado al correo electrónico y se verificará después para redirigir a un formulario que permita el cambio. Este formulario debe tener 3 campos: código, nueva contraseña y repetir contraseña. La contraseña debe ser validada con un mínimo de 8 caracteres.

Número del requisito	03
Tipo	☒ Requisito ☐ Restricción
Fuente del requisito	
Prioridad del requisito	☒ Alta/Eencial ☐ Media/Deseado ☐ Baja/ Opcional
Descripción del requisito	El sistema debe permitir crear, actualizar, listar y eliminar firmas de voz. Se requiere almacenar 3 campos: nombre, código de país, número de teléfono. Además, cada firma debe estar relacionada al usuario que inicia sesión.

Número del requisito	04
Tipo	☒ Requisito ☐ Restricción
Fuente del requisito	
Prioridad del requisito	☒ Alta/Eencial ☐ Media/Deseado ☐ Baja/ Opcional
Descripción del requisito	El sistema debe permitir hacer el registro/enrolamiento de huellas de voz en Phonexia por usuario. Las llamadas de registro tendrán un

	límite de 1 minuto con 10 segundos. El usuario tendrá la posibilidad de administrar las huellas de voz demo. El estado del registro debe almacenarse en un campo llamado is_enroll como campo booleano para la posterior verificación de la firma de voz.
--	--

Número del requisito	05
Tipo	☒ Requisito ☐ Restricción
Fuente del requisito	
Prioridad del requisito	☒ Alta/Esencial ☐ Media/Deseado ☐ Baja/ Opcional
Descripción del requisito	El sistema debe contar con la función de realizar la muestra de verificación. Debe realizar una llamada al usuario y debe mostrar por medio de una gráfica el estado de la verificación (verificado/no verificado). Además, se debe guardar en base de datos el estado en el que se encuentra la firma de voz.

Número del requisito	06
Tipo	☒ Requisito ☐ Restricción
Fuente del requisito	
Prioridad del requisito	☒ Alta/Esencial ☐ Media/Deseado ☐ Baja/ Opcional
Descripción del requisito	El sistema debe contar con una base de datos de textos de recomendación a mostrar al momento de la realización de las llamadas. Se deben listar estos textos de forma aleatoria para cada llamada. Estos textos tienen el objetivo de dar una idea a los usuarios sobre qué hablar durante la llamada y obtener huellas de voz claras y precisas.

Anexo 3. Documentación técnica

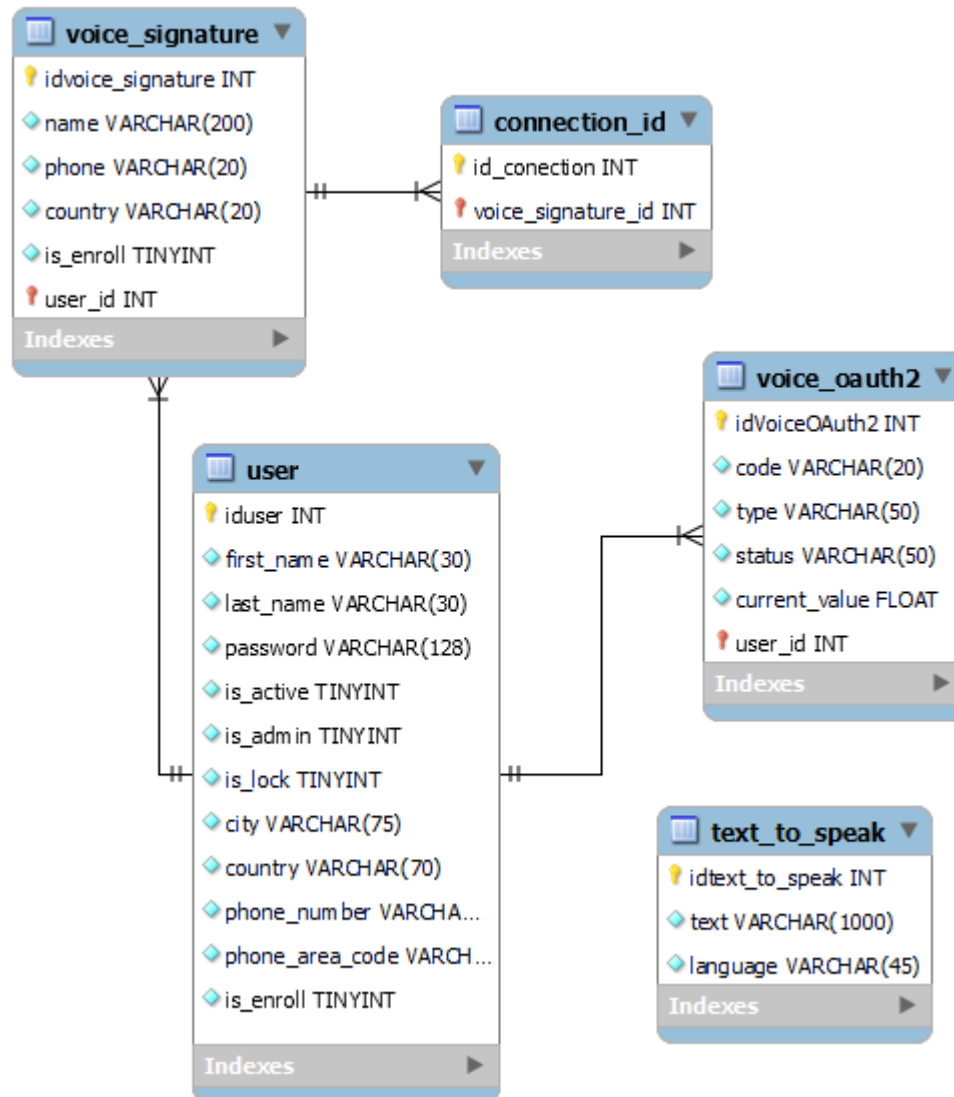
DOCUMENTACIÓN TÉCNICA

Geepy Voice

1. CAPA DE PERSISTENCIA

1.1. Modelo relacional

Figura 1. Modelo relacional



Fuente: Autor

El modelo relacional de este proyecto cuenta con muy pocas tablas ya que la licencia adquirida de Phonexia proporcionó la base de datos que almacena las huellas de voz de todos los usuarios.

En este modelo relacional contamos con 5 entidades, cuya funcionalidad principal es la siguiente:

- **Voice signature:** almacena la información suministrada por cada usuario para su huella de voz, por ejemplo: número de teléfono y código de área.
- **Connection id:** almacena las conexiones temporales generadas por aws (Amazon Web Services) que permiten la correcta transmisión del flujo de datos entre la capa de presentación y negocio.
- **Voice oauth2:** almacena el historial del proceso de cada usuario, por ejemplo: el estado en el que se encuentra la llamada (verify, enroll) y la cantidad de paquetes recibidos (estos indican segundo a segundo el estado del proceso).
- **User:** almacena toda la información relevante de cada usuario dentro de toda la aplicación de Geepy Cloud Connect.
- **Text to speak:** almacena diferentes recomendaciones de texto en forma de pregunta, por ejemplo, **¿Qué te apasiona en la vida?** Estas recomendaciones son para mostrar en pantalla durante la llamada, y son consultadas aleatoriamente con el fin de que los usuarios no permanezcan en silencio durante la llamada y tengan una idea sobre qué decir en el transcurso de la llamada para que de esta manera se pueda obtener un mejor registro de la voz.

Anexo 4. Informe de testeo

INFORME DE TESTEO

Geepy Voice

INTRODUCCIÓN

En el presente documento se encuentra el plan de pruebas realizado al software Geepy Voice durante su ciclo de pruebas de funcionalidad. La finalidad de este documento es encontrar hallazgos de posibles fallas en el aplicativo para dar su respectiva solución con el equipo de desarrollo.

METODOLOGÍA DE PRUEBAS

Para realizar las pruebas se usarán dos tipos de pruebas:

1. **Pruebas de caja negra:** Se centran en probar la funcionalidad de la aplicación sin conocer su estructura interna. Los probadores prueban la interfaz de usuario y las entradas y salidas esperadas, sin preocuparse por cómo se implementa internamente el software.
2. **Pruebas de usabilidad:** Se centran en evaluar la facilidad de uso y la experiencia del usuario de la aplicación.

ENTORNO DE PRUEBAS

Las pruebas realizadas fueron en diferentes tipos de navegadores para comprobar comportamiento de interfaz según cada uno de ellos. Se comprobó que el demo responde correctamente en cada uno de ellos:

1. Google Chrome
2. Opera
3. Opera gx
4. Firefox
5. Edge

RESULTADOS DE LAS PRUEBAS

A continuación, se adjuntan pruebas de archivo con las incidencias encontradas durante el testeó.

Tabla 1. Incidentes

Consecutivo	Módulo	Componente	Descripción	Solucionado
1	Principal	Verificación	Cuando se realiza la verificación el icono no cambia a verde	ok
2	Huella de voz	Formulario agregar huella de voz	Agregar todos los países	ok
3	Inscripción	Contador	Contador queda en 00:00 durante la llamada	ok
4	Inscripción	Barra de progreso de la llamada	Fallo en cálculo, barra sobrepasa el 100%	ok
5	Inscripción	Finalización de llamada	No redirige a pantalla principal cuando la llamada finaliza	ok
6	Pantalla principal	Sin componentes de Geepy	Agregar plataformas de Geepy	ok
7	Verificación	Visualización	Gráfico no muestra movimiento	ok

Fuente: Autor

CONCLUSIONES

En el proyecto se encontraron pocos fallos, principalmente en capa de persistencia e interfaz. Se pudo observar que el software cuenta con muy buenas métricas de calidad tanto en rendimiento y usabilidad.

7. REFERENCIAS

- IBM. (2023). Obtenido de Websocket:
<https://www.ibm.com/docs/es/was/9.0.5?topic=applications-websocket>
- Internet Security Auditors. (2005). *Inseguridad de los sistemas de autenticación en aplicaciones web. Conferencias FIST*. Obtenido de
<https://es.slideshare.net/ISecAuditors/inseguridad-de-los-sistemas-de-autenticacionaplicacionesweb>
- Likecomtic - Arquitectos de Software. (2020). *Likecomtic - Arquitectos de Software*. Obtenido de
<https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.youtube.com%2Fwatch%3Fv%3DQ5Qq5fYPpiY&psig=AOvVaw1sRxdZTPPxgr1TFs6Rxa9&ust=1685920941497000&source=images&cd=vfe&ved=0CBEQjRxqFwoTCKCT34mfqP8CFQAAAAAdAAAAABAD>
- Oracle. (2023). *MySQL Workbench*. Obtenido de
<https://www.mysql.com/products/workbench/>
- Pressman, R. (2010). *Ingeniería del Software. Un enfoque práctico*. USA: McGraw-Hill.
- Red Hat. (2023). Obtenido de Webhook:
<https://www.redhat.com/es/topics/automation/what-is-a-webhook#:~:text=Un%20webhook%20es%20una%20función,mediante%20eventos%20de%20forma%20ligera.>
- Sepúlveda Marín, J. C. (2022). *Análisis de la efectividad de los modelos de autenticación 2FA y MFA de acuerdo a los algoritmos y protocolos aplicados en la seguridad de cuentas de servicios y plataformas online en Colombia*.