



**UNIVERSIDAD DE LAS FUERZAS ARMADAS
“ESPE”**

**DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN**

ARQUITECTURA DE SOFTWARE

PARCIAL III

NRC: 3895

TEMA:

Generador de cuentos con conversión a voz usando APIs Públicas

INTEGRANTES:

**Milton Roberto Cuji Revelo
Heidy Magaly Ñacata Iza
Navarro Zambrano Johny Nicolay**

10 DE FEBRERO DEL 2025

Introducción

El avance de las tecnologías de inteligencia artificial (IA) y la creciente disponibilidad de APIs públicas han revolucionado la forma en que interactuamos con aplicaciones web, especialmente en ámbitos educativos y literarios. En este contexto, surge la necesidad de desarrollar soluciones innovadoras que combinen la generación de contenido personalizado con herramientas de accesibilidad, como la conversión de texto a voz. Este proyecto propone la creación de una aplicación web interactiva que permita a los usuarios generar cuentos personalizados mediante IA y escucharlos en tiempo real, utilizando tecnologías modernas y APIs públicas.

La aplicación está diseñada como una solución fullstack, donde el backend se implementa con Django, un framework de Python conocido por su escalabilidad y facilidad de desarrollo de APIs RESTful. Por otro lado, el frontend se construye con React, una biblioteca de JavaScript que permite crear interfaces de usuario dinámicas y reactivas. Para el almacenamiento de datos, se utiliza PostgreSQL, una base de datos relacional robusta y confiable, ideal para gestionar información estructurada como cuentos generados y archivos de audio. Además, se integra la API de IEventLabs para la generación de cuentos personalizados mediante IA, lo que permite ofrecer una experiencia única y altamente personalizable para los usuarios.

Este proyecto no solo busca demostrar el potencial de las tecnologías modernas en la creación de aplicaciones interactivas, sino también fomentar el acceso a la literatura de manera inclusiva. Al combinar la generación de contenido con IA y la conversión de texto a voz, la aplicación se convierte en una herramienta valiosa para docentes, padres de familia y personas con discapacidad visual, entre otros. La arquitectura del sistema, basada en Django, React, PostgreSQL y IEventLabs, garantiza un rendimiento óptimo y una experiencia de usuario fluida, sentando las bases para futuras mejoras y expansiones.

Objetivo General

Desarrollar una aplicación web interactiva que permita la generación de cuentos personalizados mediante inteligencia artificial (IA) y su conversión a voz en tiempo real, utilizando tecnologías modernas como Django, React, PostgreSQL y APIs públicas como IEventLabs.

Objetivos Específicos

Implementar un backend con Django:

- Crear una API RESTful para gestionar las solicitudes del frontend y comunicarse con las APIs externas.
- Asegurar la escalabilidad y el rendimiento del sistema.

Desarrollar una interfaz de usuario interactiva con React:

- Diseñar una interfaz amigable y accesible que permita a los usuarios personalizar cuentos y escuchar narraciones en tiempo real.
- Integrar el frontend con el backend mediante solicitudes HTTP.

Integrar la API de IEventLabs para la generación de cuentos:

- Utilizar la API de IEventLabs para generar cuentos personalizados basados en parámetros proporcionados por el usuario.
- Garantizar que los cuentos generados sean originales y adaptados a las preferencias del usuario.

Implementar una base de datos con PostgreSQL:

- Almacenar temporalmente cuentos generados y archivos de audio para su posterior recuperación.
- Optimizar el rendimiento de la aplicación mediante el uso de una base de datos relacional.

Optimizar el rendimiento y la escalabilidad:

- Utilizar un API Gateway para centralizar y balancear las solicitudes a las APIs externas.
- Gestionar solicitudes concurrentes de manera eficiente.

Análisis y Planificación:

Se definieron los requisitos del sistema, incluyendo las funcionalidades principales: generación de cuentos, conversión a voz y una interfaz de usuario interactiva.

Se seleccionaron las siguientes tecnologías a utilizar:

Django para el backend,

React para el frontend,

PostgreSQL para la base de datos y la API de IEventLabs para la generación de cuentos.

Desarrollo del Backend (Django):

- Se implementó una API RESTful utilizando Django Rest Framework (DRF) para gestionar las solicitudes del frontend.
- Se integró la API de IEventLabs para la generación de cuentos personalizados, enviando parámetros como tema, protagonista y estilo.
- Se configuró PostgreSQL como base de datos para almacenar temporalmente los cuentos generados y los archivos de audio.
- Se implementó la lógica para gestionar solicitudes concurrentes y optimizar el rendimiento del sistema.

Desarrollo del Frontend (React):

- Se diseñó una interfaz de usuario interactiva y amigable utilizando React.
- Se implementaron formularios para que los usuarios ingresen parámetros de personalización (tema, protagonista, estilo, etc.).
- Se integró el frontend con el backend mediante solicitudes HTTP (utilizando fetch o axios).
- Se agregaron componentes para mostrar los cuentos generados y permitir la selección de voces para la narración.

Integración de APIs Externas:

- Se configuró un API Gateway para centralizar y balancear las solicitudes a las APIs externas (IEventLabs y APIs de texto a voz).
- Se implementó la conversión de texto a voz utilizando APIs como Google TTS o Amazon Polly, permitiendo a los usuarios escuchar los cuentos en tiempo real.

Pruebas y Validación:

- Se realizaron pruebas unitarias en el backend para verificar el correcto funcionamiento de los endpoints y la lógica de negocio.
- Se realizaron pruebas de integración para asegurar que el frontend y el backend se comunicaran correctamente.
- Se validó la escalabilidad del sistema mediante pruebas de carga, simulando múltiples usuarios concurrentes.

Entorno de Desarrollo:

El proyecto se desarrolló y ejecutó en un entorno local, utilizando herramientas como Docker para gestionar los contenedores de la base de datos (PostgreSQL) y el servidor de Django.

El frontend se ejecutó en un servidor de desarrollo local utilizando React Scripts.

No se realizó un despliegue en la nube, ya que el foco del proyecto fue el desarrollo y la validación de las funcionalidades en un entorno controlado.

Resultados:

Funcionalidades Implementadas

Generación de Cuentos Personalizados:

- Se logró integrar la API de IEventLabs para generar cuentos originales basados en parámetros como tema, protagonista y estilo.
- Los cuentos generados son únicos y adaptados a las preferencias del usuario, lo que permite una experiencia personalizada.

Conversión de Texto a Voz:

- Se implementó la conversión de texto a voz en tiempo real utilizando APIs como Google TTS o Amazon Polly.
- Los usuarios pueden seleccionar entre diferentes voces y estilos de narración, lo que enriquece la experiencia auditiva.

Interfaz de Usuario:

- La interfaz desarrollada con React es intuitiva y accesible, permitiendo a los usuarios interactuar fácilmente con la aplicación.
- Los usuarios pueden ingresar parámetros de personalización, generar cuentos y escuchar las narraciones en tiempo real.

Base de datos:

- Se configuró PostgreSQL para almacenar temporalmente los cuentos generados y los archivos de audio.
- Esto permite a los usuarios recuperar cuentos y narraciones previamente generados, mejorando la eficiencia del sistema.

API Gateway:

- Se implementó un API Gateway para centralizar y balancear las solicitudes a las APIs externas.
- Esto optimiza el rendimiento del sistema y permite gestionar múltiples solicitudes concurrentes de manera eficiente.

Pruebas y Validación

Pruebas Unitarias:

- Se realizaron pruebas unitarias en el backend para verificar el correcto funcionamiento de los endpoints y la lógica de negocio.
- Todas las pruebas unitarias fueron exitosas, asegurando que cada componente del backend funciona como se espera.

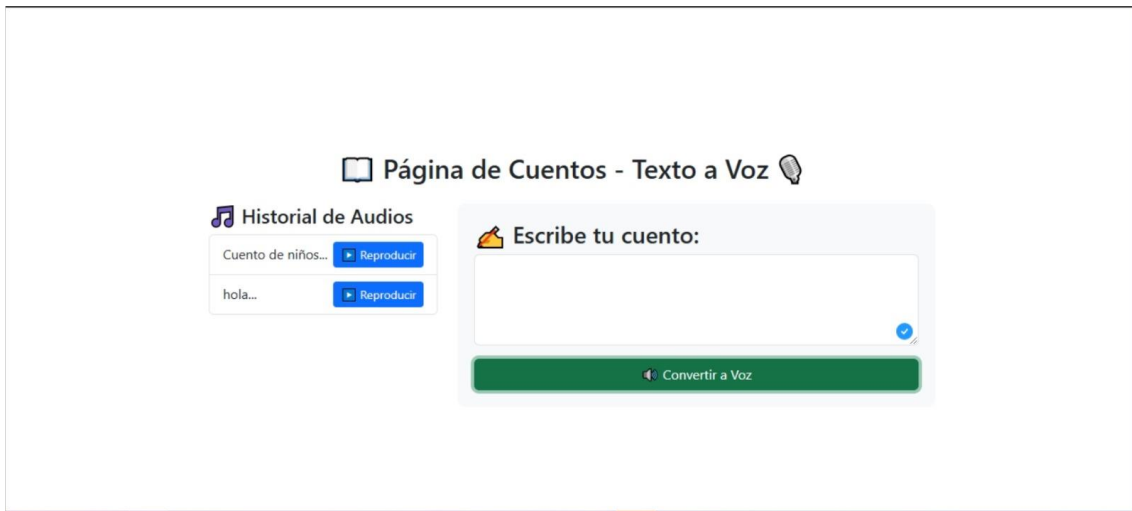
Pruebas de Integración:

- Se realizaron pruebas de integración para asegurar que el frontend y el backend se comunicaran correctamente.
- Se validó que las solicitudes del frontend se procesan adecuadamente en el backend y que las respuestas se muestran correctamente en la interfaz de usuario.

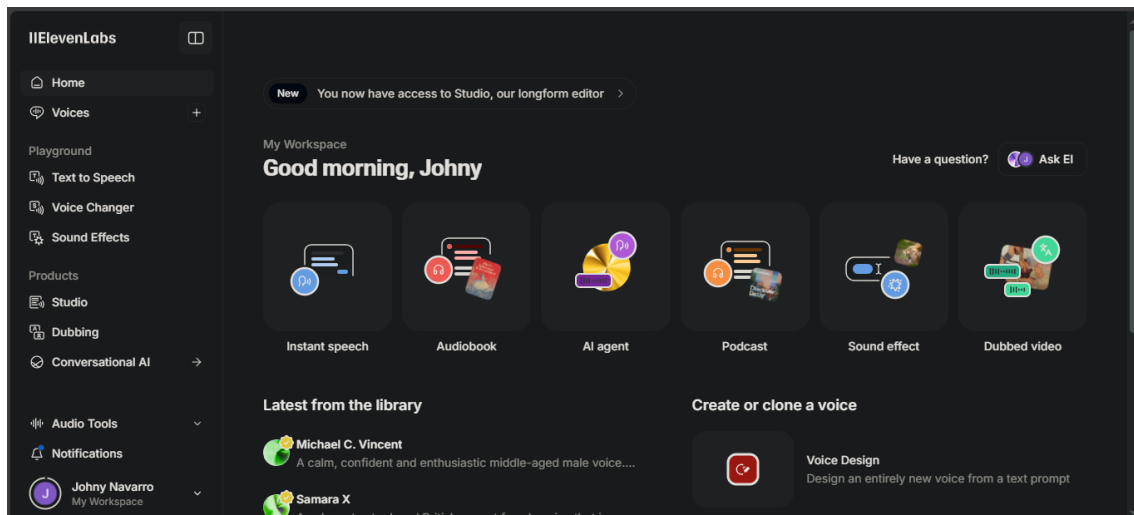
Capturas de la integración de componentes

Página principal

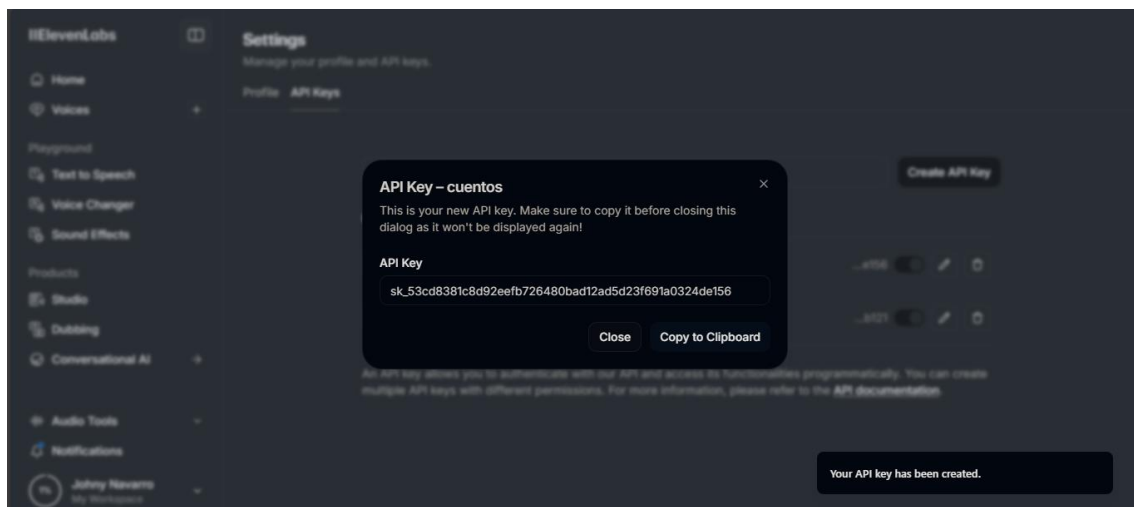
Front-end React



Pantalla principal donde se obtiene la api gratuita



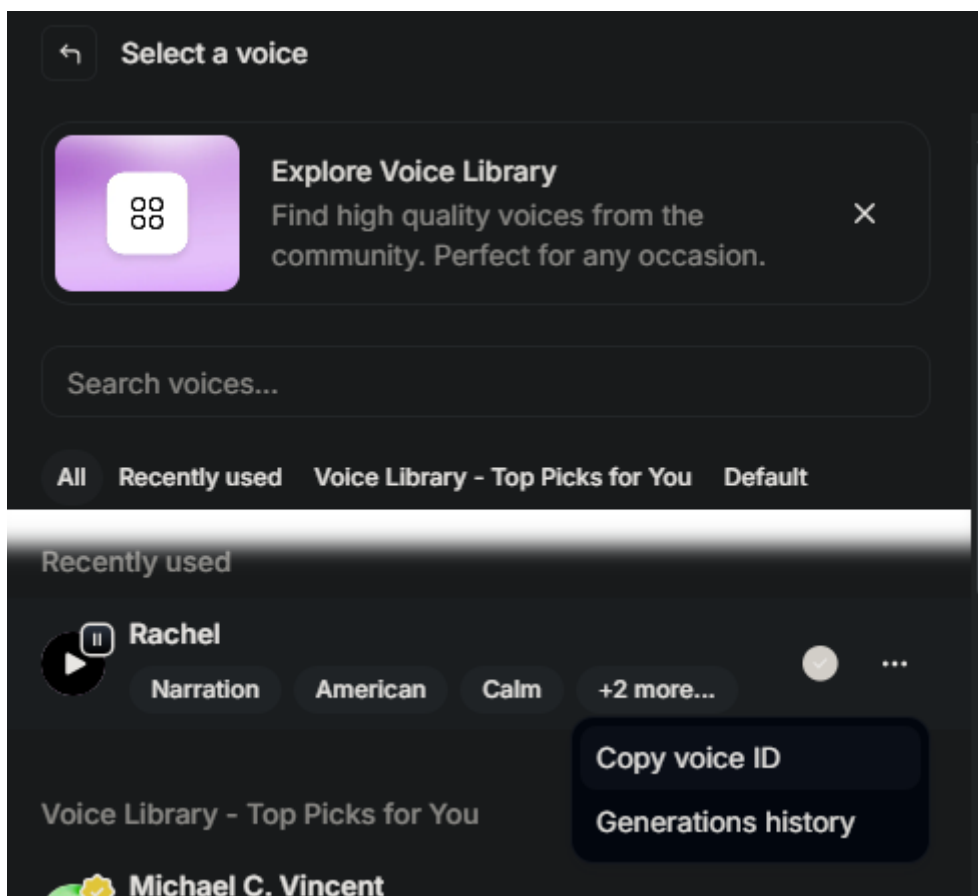
Creación y obtención del api



Instalación de librerías para el back-end

- Instalación de librerías (se instala la version de python .3.11)
- pip install pipwin
- pipwin install pyaudio
- Se reemplaza por la ID voz y api:

```
import pygame
API_KEY = "sk_4cd0e0205b906668054976692bdc59a79110cb358400b121"
VOICE_ID = "21m00Tcm4TlvDq8ikwAM"
def text_to_speech(text):
```



Conclusiones

El desarrollo de esta aplicación web interactiva para la generación de cuentos personalizados y su conversión a voz en tiempo real ha demostrado el potencial de combinar tecnologías modernas con APIs públicas para crear soluciones innovadoras y accesibles.

A lo largo del proyecto, se lograron los objetivos planteados, implementando un sistema robusto y escalable que ofrece una experiencia única a los usuarios.

- Se logró combinar eficientemente Django (backend), React (frontend), PostgreSQL (base de datos) y APIs públicas como IIEventLabs y Google TTS,

demostrando que estas tecnologías son compatibles y potentes para el desarrollo de aplicaciones web fullstack.

- La API de IEventLabs permitió crear cuentos únicos y adaptados a las preferencias del usuario, lo que enriqueció la experiencia interactiva y abrió posibilidades para su uso en ámbitos educativos y literarios.
- La integración de Google TTS proporcionó una conversión fluida y en tiempo real de los cuentos generados, ofreciendo a los usuarios la posibilidad de escuchar narraciones con diferentes voces y estilos.
- La interfaz desarrollada con React resultó ser amigable y fácil de usar, permitiendo a los usuarios generar cuentos y escuchar narraciones sin complicaciones, lo que mejora la experiencia de usuario.

Bibliografía

- <https://docs.djangoproject.com/>
- <https://reactjs.org/docs/getting-started.html>
- <https://www.postgresql.org/docs/>
- <https://cloud.google.com/text-to-speech/docs>
- <https://docs.docker.com/>
- <https://www.django-rest-framework.org/>
- <https://reactrouter.com/>