

EARFIT: Aplicación Para Entrenamiento Auditivo Musical Basada en Next.js y TypeScript

Trabajo fin de grado

Grado en Ingeniería Informática

Autor: *Alberto Gómez Cano*

Tutor: *Manuel Rubio Sánchez*



Universidad
Rey Juan Carlos

Escuela Técnica Superior
Ingeniería Informática

Contenido



1 Introducción

2 Metodologías

- Design Thinking
- Lean Startup
- Scrum
- DevOps

3 Desarrollo

- Tecnologías
- Detalles de Implementación
- Progressive Web App
- Software QA

4 Conclusiones



Entrenamiento Auditivo

¿Qué es?

- Identificar y asociar los elementos musicales con el sonido.
- Los músicos, productores y DJs pueden beneficiarse.
- Permite sacar canciones más rápido, improvisar mejor y llevar al instrumento las melodías.
- Los ejercicios más comunes incluyen identificar notas, intervalos, escalas...



Objetivos

- **Objetivo principal:**

- Crear una aplicación que permita a músicos desarrollar su oído musical mediante el entrenamiento auditivo.

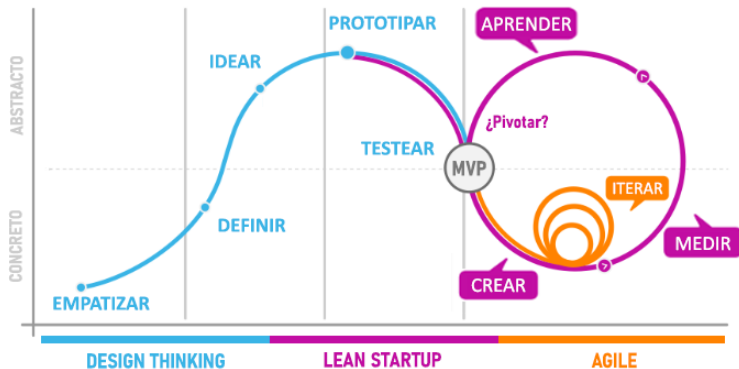
- **Subobjetivos:**

- Desarrollar una interfaz interactiva.
- Implementar diferentes tipos de ejercicio personalizables.
- Incluir varios instrumentos para practicar con sus sonidos.



Metodologías

Design Thinking, Lean Startup, Scrum y DevOps

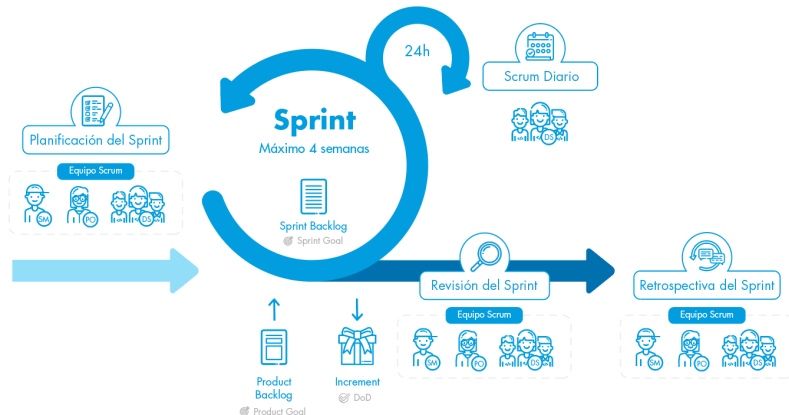


- **Técnicas:** User Persona, MindMap, MoSCoW, Mobile First...



Scrum

Proceso de Gestión del Desarrollo Ágil

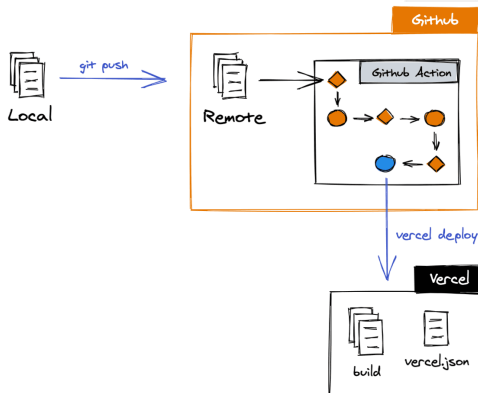


- **Artefactos:** User Story Map, Scrum Board y User Stories.



DevOps

Agilizar los Procesos del Entorno de Desarrollo al de Producción



- **Integración Continua (CI):** Git, GitHub y GitFlow.
- **Despliegue Continuo (CD):** GitHub Actions, Vercel y DPS.



Stack Tecnológico

NEXT.js



TS

node.js

▲ Vercel

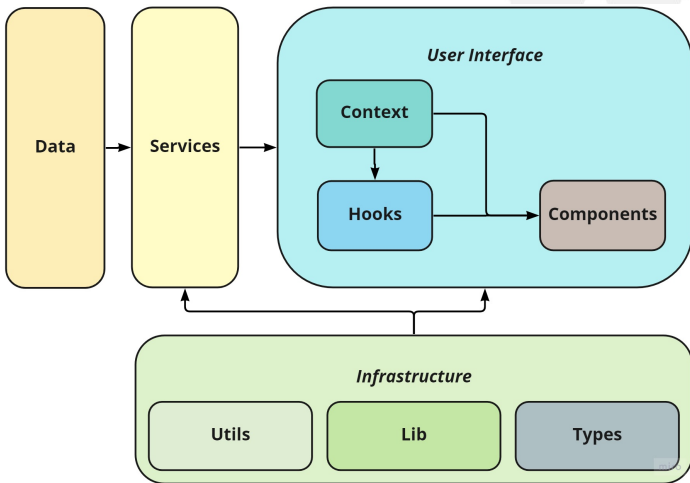


- **Next.js:** Enrutamiento basado en páginas, Prerendering, Code Splitting, Prefetching, Fast Refresh, SWC y WebPack...
- **React:** Componentes, DOM Virtual, Sintaxis JSX...
- **Typescript:** Tipos estáticos para JavaScript.
- **Node.js:** Entorno de ejecución, librerías (NPM vs. Yarn).
- **VScode:** Git integrado, extensiones (ESLint y Prettier).



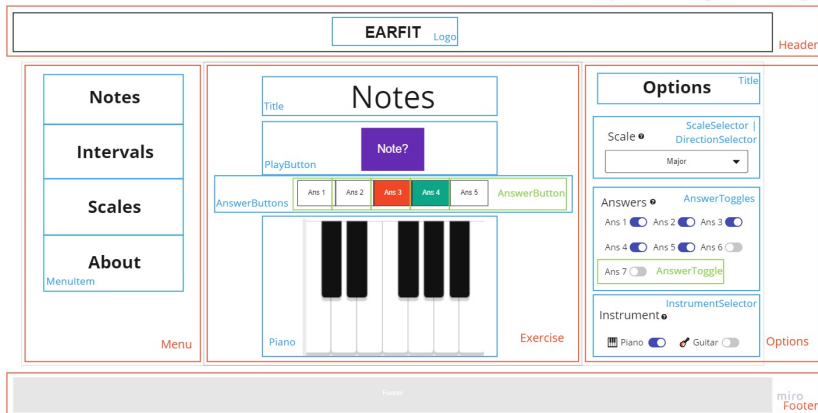
Arquitectura

Estructura, Funcionamiento e Interacción



Componentes

Renderizan la Interfaz de Usuario



Hooks y Context

Encapsulan la Lógica de Estado

- **useExercise.tsx**: Establecer las respuestas para el ejercicio.
- **useAnswerToggles.tsx**: Añadir y quitar respuestas a la pregunta.
- **useAnswerButtons.tsx**: La lógica de los botones de respuesta.
- **useAnswer.tsx**: Calcular la respuesta a preguntar.
- **usePlayButton.tsx**: Reproducir la respuesta.
- **EarfitContext.tsx**: Establecer y seleccionar los instrumentos.
Se usa con el Hook `useInstrumentContext()`.



Servicios

Proveen los Datos

- **instrumentService.ts**: Provee los instrumentos de la aplicación.
- **noteService.ts**: Provee las respuestas correspondientes al ejercicio de notas.
- **intervalService.ts**: Provee las respuestas correspondientes al ejercicio de intervalos.
- **scaleService.ts**: Provee las respuestas correspondientes al ejercicio de escalas.



Infraestructura

Apoyan a los Servicios y la Interfaz

Tipos más importantes:

- **Instrument:** Id, noteplayer, nombre para mostrar.
- **Answer:** Id, array de notas, nombre para mostrar.

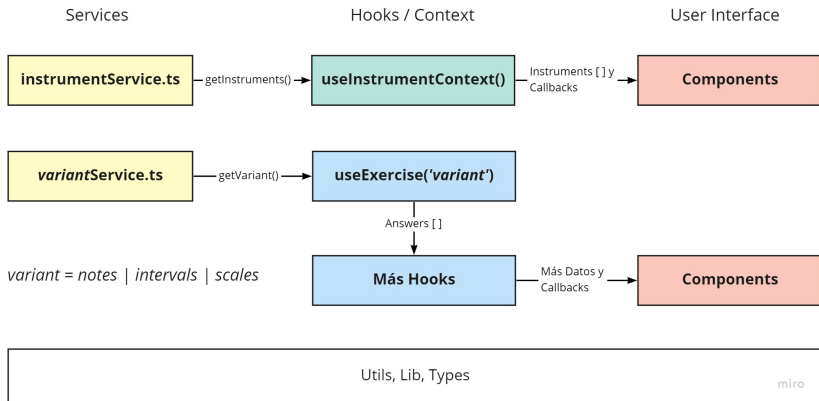
Librerías más importantes:

- **Tonal.js:** Manipular elementos musicales.
- **Soundfont-player:** Cargar y reproducir archivos MIDI.js.
- **Soundfont-wrapper:** Simplificar “soundfont-player”.
Crear los Noteplayer para cada instrumento.



Comportamiento de una Página

Interacción entre Componentes, Hooks y Servicios



Progressive Web App

Instalable como App Nativa en PC, Móvil y Tablet

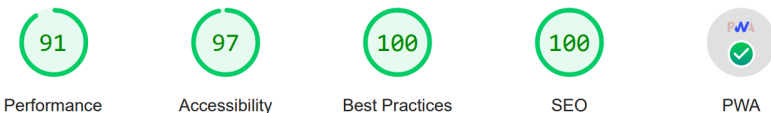
- Conexión segura HTTPS.
- Carga sin conexión, para ello tiene un Service Worker.
- Información como nombre, autor, icono y descripción en un documento JSON llamado Manifest.
- Icono de al menos 144x144 px en formato PNG.



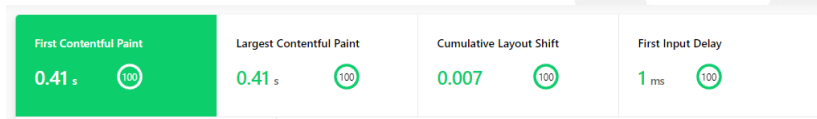
Software QA

Estándares y Calidad del Software

- **Google Lighthouse:** Auditoría de Calidad.



- **Vercel Analytics:** Experiencia de Usuario (Web Vitals).



Conclusiones

Objetivos Alcanzados:

- PWA que permite desarrollar el oído musical mediante entrenamiento auditivo.
- Diferentes tipos de ejercicios personalizables.
- Varios instrumentos para practicar con sus sonidos.
- Buenas prácticas, gestión ágil y DevOps.

Trabajos Futuros:

- Ejercicios de acordes, ritmos, progresiones, modo nocturno y varios idiomas.



Demostración



URL: <https://earfit.vercel.app/>



EARFIT: Aplicación Para Entrenamiento Auditivo Musical Basada en Next.js y TypeScript

Trabajo Fin de Grado

Grado en Ingeniería Informática – Curso 2021-2022

Autor: *Alberto Gómez Cano*

Tutor: *Manuel Rubio Sánchez*



Universidad
Rey Juan Carlos

Escuela Técnica Superior
Ingeniería Informática