

ESCUELA POLITÉCNICA NACIONAL

**FACULTAD DE INGENIERÍA ELÉCTRICA Y
ELECTRÓNICA**

**DESARROLLO DE UN ADD-ON SOBRE LA PLATAFORMA
GOOGLE SPREADSHEETS PARA LA INTERPRETACIÓN DE
GRÁFICOS USANDO TÉCNICAS DE SONORIZACIÓN ENFOCADO
A PERSONAS NO VIDENTES**

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
TECNOLOGÍAS DE LA INFORMACIÓN**

BRYAN SEBASTIAN TORRES CUENCA
bryan-torres14@hotmail.com

DIRECTOR: DRA. ANA MARÍA ZAMBRANO VIZUETE
ana.zambrano@epn.edu.ec

DMQ, Febrero 2022

CERTIFICACIONES

Yo, Bryan Sebastian Torres Cuenca declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.



Bryan Sebastian Torres Cuenca

Certifico que el presente trabajo de integración curricular fue desarrollado por Bryan Sebastian Torres Cuenca, bajo mi supervisión.



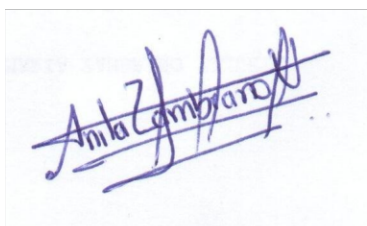
Dra. Ana María Zambrano Vizúete
DIRECTOR

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.



SR. BRYAN SEBASTIAN TORRES CUENCA



DRA. ANA MARÍA ZAMBRANO VIZUETE

DEDICATORIA

A mis padres por haberme ofrecido todas las oportunidades para lograr conseguir este objetivo, además de brindarme su apoyo incondicional en todas etapas de mi vida.

A mis hermanos, por haber formado parte de esta valiosa etapa y ofrecerme su ayuda siempre que fue necesario.

A todos mis amigos que han sido parte integral de mi desenvolvimiento personal y profesional. De manera especial a Daniela por haberme apoyado incondicionalmente durante esta travesía.

Bryan Sebastian Torres Cuenca

AGRADECIMIENTO

Brindo un agradecimiento especial a la Dra. Ana María Zambrano, directora de este trabajo y docente de la Escuela Politécnica Nacional, por darme las directrices correctas, brindarme su apoyo y permitir que este trabajo se terminara a buen recaudo.

A su vez, ofrezco mis sinceros agradecimientos al Dr. Felipe Grijalva, docente de la Escuela Politécnica Nacional, por haberme ofrecido su tutoría en varios temas del presente trabajo.

Junto a ellos agradezco a todos los docentes que formaron parte de la formación recibida por parte de la Escuela Politécnica Nacional.

Bryan Sebastian Torres Cuenca

ÍNDICE DE CONTENIDOS

CERTIFICACIONES.....	I
DECLARACIÓN DE AUTORÍA.....	II
DEDICATORIA.....	III
AGRADECIMIENTO.....	IV
ÍNDICE DE CONTENIDOS	V
RESUMEN	VII
ABSTRACT	VIII
1 INTRODUCCIÓN.....	1
1.1 OBJETIVO GENERAL	2
1.2 OBJETIVOS ESPECÍFICOS	2
1.3 ALCANCE	2
1.4 MARCO TEÓRICO.....	5
1.4.1 SONORIZACIÓN Y MANEJO DE AUDIO ESPACIAL.....	5
1.4.2 HERRAMIENTAS Y TECNOLOGÍAS PARA LA IMPLEMENTACIÓN DEL PROTOTIPO	9
1.4.3 METODOLOGÍA KANBAN	12
2 METODOLOGÍA.....	13
2.1 DISEÑO	13
2.1.1 PLANTEAMIENTO DE TABLERO KANBAN.....	13
2.1.2 DEFINICIÓN DE LA ARQUITECTURA DEL PROTOTIPO	14
2.1.3 REQUERIMIENTOS FUNCIONALES Y NO FUNCIONALES DEL PROTOTIPO	15
2.1.4 DIAGRAMA DE CASOS DE USO DEL PROTOTIPO	19
2.1.5 DIAGRAMA DE ACTIVIDADES DEL PROTOTIPO	20
2.1.6 DISEÑO DE LA INTERFAZ	21
2.2 IMPLEMENTACIÓN.....	24
2.2.1 ACTUALIZACIÓN DEL TABLERO KANBAN	24
2.2.2 CODIFICACIÓN DEL DIAGRAMA DE ACTIVIDADES	25
2.2.3 CODIFICACIÓN DE LA INTERFAZ	39
3 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES.....	43
3.1 RESULTADOS.....	43
3.1.1 ACTUALIZACIÓN DE TABLERO KANBAN	43
3.1.2 DEFINICIÓN DE ENTORNO DE PRUEBAS.....	44

3.1.3	VALIDACIÓN DE REQUERIMIENTOS FUNCIONALES.....	45
3.1.4	VALIDACIÓN DE REQUERIMIENTOS NO FUNCIONALES	47
3.1.5	CIERRE DEL TABLERO KANBAN	51
3.2	CONCLUSIONES Y RECOMENDACIONES	52
3.2.1	CONCLUSIONES.....	52
3.2.2	RECOMENDACIONES.....	54
4	REFERENCIAS BIBLIOGRÁFICAS	56
5	ANEXOS.....	60

RESUMEN

La sonorización o sonificación es una técnica que consiste en la generación de interpretaciones auditivas a partir de datos, ya sean, textos, tablas, imágenes o cualquier otro elemento multimedia. La mencionada técnica puede ser implementada mediante diversos métodos, dependiendo del tipo de contenido a sonificar.

En el presente Trabajo de Integración Curricular se planteó el desarrollo de un *add-on* que permita la generación de interpretaciones auditivas, equivalentes a gráficos tradicionales generados por hojas de cálculo; en este caso *Google Spreadsheets*. Con el fin de facilitar la compatibilidad del prototipo se utilizó una técnica de sonorización basada en señales sinusoidales, complementado con el uso de audio espacial; todo esto totalmente programado sobre *googlescript*, *javascript* y *HTML5*, sobre la plataforma *Apps Script*.

Durante el primer capítulo se exploran los pormenores acerca de la sonorización de datos basada en señales sinusoidales y como el correcto manejo de sus parámetros asociados puede otorgar interpretaciones fieles a la realidad, a su vez, se destaca la espacialización del audio como una técnica complementaria, que permite emular el comportamiento del audio en la naturaleza. La teoría sobre sonificación se sustenta a su vez sobre las tecnologías que permiten realizar estas tareas sobre plataformas web, además de una descripción acerca de su integración con *Google Spreadsheets* y la arquitectura detrás de esta herramienta.

A lo largo del segundo capítulo se definen los patrones de diseños seguidos para alcanzar una interfaz adaptada correctamente a usuarios no videntes, procurando seguir las recomendaciones planteadas por la WCAG (*Web Content Accessibility Guidelines*) [1]. Se detallan a su vez los pasos a seguir para implementar el prototipo sobre la arquitectura *Apps Script*, las funcionalidades que fueron implementadas y la relación latente que manejan las mismas con la interfaz final de usuario.

Finalmente, se pone a prueba el prototipo desarrollado en un entorno de pruebas controlado. Se verifica el desenvolvimiento del *add-on* en un entorno real mediante el uso del mismo por usuarios que simulan ser individuos no videntes y se realizan las correcciones obtenidas mediante la retroalimentación de los usuarios de prueba.

PALABRAS CLAVE: Sonorización, audio espacial, Web Speech API, Web Audio API, Google Apps Script, Google Spreadsheets.

ABSTRACT

The sonorization or sonification is a technique that consists in the generation of audio interpretations from data, whether they are texts, tables, images or any other multimedia object. The mentioned technique can be implemented through various methods, depending on what kind content is going to be sonificated.

In this Curricular Integration Paper, the development of an *add-on* that allows the generation of auditory interpretations, equivalent to traditional graphics generated by spreadsheets, is proposed, in this case *Google Spreadsheets*. In order to facilitate the compatibility of the prototype, a sonification technique based on sinusoidal waves was used, complemented with spatial audio; all this programmed on *googlescript*, *javascript* and *HTML5*, over *Apps Script* platform.

During first chapter the details about data sonification based on sinusoidal waves and how the right management of their associated parameters can grant interpretations faithful to reality are explored, in turn, audio spatialization is highlighted as a complementary technique, that allows emulate the behavior of sound in nature. The sonification theory is supported on technologies that allow these tasks to be carried out on web platforms, as well as a description of its integration with *Google Spreadsheets* and the architecture behind this tool.

Throughout the second chapter the design patterns followed to achieve an adapted interface correctly integrated to blind users are defined, trying to follow the recommendations proposed by WCAG (*Web Content Accessibility Guidelines*). Simultaneously, the steps to implement the prototype over the Apps Script architecture are detailed, functionalities that were implemented and the relationship that they manage with the final user interface.

Finally, the developed prototype is tested in a controlled test environment. The correct functionality of the *add-on* is verified in a real environment through its use by users who pretend to be blind users, the feedback given by test users was used to make final corrections.

KEYWORDS: Sonorization, Spatial Audio, Web Speech API, Web Audio API, Google Apps Script, Google Spreadsheets.

1 INTRODUCCIÓN

La utilización de gráficos permite visualizar y establecer relaciones entre los datos imposibles de conseguir exclusivamente mediante el análisis textual de la información. El manejo de herramientas de ofimática como hojas de cálculo como herramienta para manejar datos relativamente complejos y a su vez interpretarlos mediante gráficos y diversos tipos de funciones matemáticas, puede significar la diferencia entre embonar y no en el mundo contemporáneo.

Las herramientas actuales que permiten gestión de hojas de cálculo presentan pocas opciones de accesibilidad para personas no videntes y están enfocadas a un uso básico, mediante lectores de pantalla o permitiendo que lectores de pantalla externos puedan leer el contenido de la aplicación. La sonorización permite generar interpretaciones auditivas de prácticamente cualquier tipo de contenido, obteniendo resultados exitosos dependiendo del tipo de sonificación, los datos y la población objetivo.

Herramientas como la *Web Audio API* [2] y la *Web Speech API* [3] permiten que sea viable la sonorización embebida en servicios web, no necesariamente usando exclusivamente síntesis de texto para describir el entorno, sino que permitiendo al usuario participar de la creación de herramientas de análisis de datos del tipo de gráficos y a su vez compartirlos con sus colaboradores o personal interesado. *Google* mediante su plataforma *Apps Script* permite la expansión de las funcionalidades de sus servicios web de ofimática y otro tipo de herramientas. *Google Spreadsheets*¹ permite la creación de funcionalidades totalmente nuevas, mediante la creación de *add-ons*, mismos que deberán ser programados sobre *googlescript*, *javascript* y *HTML*.

Crear un entorno integrado, que permita almacenar, manejar, compartir e interpretar datos es posible; adicionalmente se puede conseguir un procesamiento prácticamente nulo por parte de usuarios y desarrolladores. *Google Spreadsheets* conjuntamente con varios servicios y funcionalidades propias de *Google*, mediante un desarrollo enfocado en la accesibilidad, crea la posibilidad de desarrollar una herramienta de sonificación eficaz, que funcione como servicio web y tenga la suficiente compatibilidad como para ser usado por una buena parte de la población ya sean usuarios con o sin algún tipo de discapacidad visual.

A lo largo del presente Trabajo de Integración Curricular se explora el desarrollo de un *add-on* sobre *Google Spreadsheets* que permita realizar sonorizaciones correctas, a partir

¹ *Google Spreadsheets* es una herramienta de gestión de hojas de cálculo enfocada al uso web compartido y la compatibilidad con otras herramientas comerciales como *Microsoft Excel* [10].

de los datos disponibles en una hoja de cálculo y el acceso a un navegador web actualizado, sin la necesidad de grandes inversiones o instalaciones exageradamente complejas.

1.1 OBJETIVO GENERAL

Desarrollar un *add-on* sobre la plataforma *Google Spreadsheets* para la interpretación de gráficos usando técnicas de sonificación enfocado a personas no videntes.

1.2 OBJETIVOS ESPECÍFICOS

1. Analizar las tecnologías necesarias para la implementación del *add-on*.
2. Diseñar el *add-on* según las actividades a realizar.
3. Implementar las funciones del *add-on*.
4. Analizar los resultados obtenidos en escenarios de prueba.

1.3 ALCANCE

El presente Trabajo de Integración Curricular propone el desarrollo de un *add-on* para ser utilizado en la herramienta de gestión de hojas de cálculo *Google Spreadsheets*. La herramienta por desarrollar tendrá la función de ayudar a usuarios con discapacidad visual a seleccionar los datos que requieran, seleccionar entre cuatro tipos de gráficos (gráficos de barras, líneas, pastel y dispersión), graficar los datos y proporcionar una descripción auditiva mediante una sonificación de los datos, misma que incluirá audio espacial. De esta manera, el usuario no vidente podrá “entender” el comportamiento de los datos de manera sencilla en función de sus capacidades.

La herramienta se desarrollará sobre el lenguaje *googlescript (Google App Script)*, lenguaje basado en *javascript* que cuenta además de las funciones básicas con módulos y funcionalidades propias relacionadas a los servicios de *Google*. Todo el procesamiento del *add-on* será realizado mediante *Google Cloud* en los servidores propios de la empresa y el cliente únicamente consumirá el *add-on* mediante la interfaz gráfica que será programada como una *API* web tradicional sobre *HTML5*.

Para ofrecer todas las funcionalidades planteadas el *add-on* requerirá de la realización de dos acciones previas:

- Utilización de algún lector de pantalla para facilitar el proceso de instalación y ejecución del *add-on*. En el caso puntual se plantea el uso de *Google ChromeVox*².
- Codificación sobre *App Scripts* en la herramienta *Google Spreadsheets*, para posteriormente ejecutarlo con el fin de asegurar que el *add-on* sea ejecutado y almacenado en *Google Cloud*.

Adicionalmente, se destacan las funciones que realizará el *add-on*:

a) Selección e inserción de datos:

El usuario no vidente será capaz de ingresar datos en una hoja de cálculo, pudiendo o no apoyarse en algún lector de pantalla. Tras finalizar la inserción de datos se seleccionarán los mismos utilizando el teclado del computador, el ratón o ingresando un rango de celdas. El *add-on* dará una retroalimentación auditiva acerca de la carga correcta de los datos.

b) Selección del tipo de gráfico:

El usuario no vidente tendrá la posibilidad de seleccionar entre cuatro distintos tipos de gráficos citados posteriormente. Al seleccionar el tipo de gráfico se generará una retroalimentación auditiva para permitir que el individuo sea capaz de entender y verificar que está realizando lo que requiere. El usuario deberá ingresar previamente los datos a ser graficados. Al seleccionar el tipo de gráfico en el *add-on* se deberá confirmar la decisión. Los gráficos disponibles son:

- Gráfico de barras.
- Gráfico de líneas.
- Gráfico de pastel.
- Gráfico de dispersión.

c) Modificación de parámetros del gráfico:

Los parámetros del gráfico serán tomados directamente de los datos disponibles en las celdas de la hoja de cálculo, de modo que el usuario con discapacidad visual no tenga la necesidad de realizar inserciones extras, que a su vez puedan resultar complejas. Es

² *Google ChromeVox*, también conocido como *Screen Reader*, es un complemento para *Google Chrome*. Permite dotar al navegador de la función de lectura de pantalla en sitios web [41].

así como, en caso de estar ausentes parámetros como el título del gráfico, este será rellanado automáticamente por un valor predeterminado.

d) Tratamiento de los datos:

Los datos ingresados requerirán de un tipo de tratamiento especial para poder adaptarlos a los distintos tipos de gráficos que se van a generar. Los datos extraídos de las celdas de la hoja de cálculo serán transformados a matrices, además de ser escalados y adaptados según la naturaleza propia del tipo de gráfico. Adicionalmente, estos deberán ser organizados y gestionados por el sistema según sea más conveniente.

e) Sonificación de los datos y adaptación con audio espacial:

Se deberán tratar los datos de modo que estos puedan ser transformados a señales de audio. El sistema será capaz de generar un sonido específico que se adapte al gráfico que fue seleccionado por el usuario no vidente, de modo que al reproducirse él tenga una idea correcta de cómo se comportan los datos que agregó. Se requerirá del manejo de sonido estéreo para ofrecer una idea más real del gráfico en cuestión. Adicionalmente, se deberá modificar frecuencias de sonido en rangos audibles que demuestren un cambio dependiendo del tipo de gráfico; además de tocar otros parámetros propios de señales de audio como el periodo y los armónicos según sea el caso. La señal de audio generada será capaz de ofrecer una experiencia multidimensional, con el fin de ubicar al usuario no vidente en el espacio y ofrecer una mejor interpretación de los datos. Mediante el estudio de técnicas existentes de manejo de audio, se procurará que el sonido generado no sea perjudicial para el usuario.

f) Compartir contenido con otros usuarios:

El *add-on* será capaz de generar un gráfico visualizable según el formato propio de *Google Spreadsheets*; esta funcionalidad será complementaria a la función principal anteriormente descrita. Esto permitirá no solo que el usuario no vidente sea capaz de explorar los gráficos, sino que a su vez este sea capaz de compartir los mismos con otros usuarios que no sufran de discapacidad visual.

Este Trabajo de Integración Curricular tendrá un producto final demostrable. Adicionalmente, debido a que la programación se realizará sobre uno de los servicios de *Google* se puede solicitar su distribución en la tienda de *add-ons* para *Google Spreadsheets*.

1.4 MARCO TEÓRICO

1.4.1 SONORIZACIÓN Y MANEJO DE AUDIO ESPACIAL

La sonorización o sonificación se define como la transformación de las relaciones latentes en los datos hacia relaciones perceptibles mediante parámetros acústicos con el fin de facilitar la comunicación e interpretación de la información [4]. De esta manera, se puede migrar la información a ambientes completamente distintos, con respecto a su fuente. El empleo de esta técnica es complejo; esto se debe a que los ejes referenciales manejados en la información de fuente deben ser migrados hacia nuevos ejes que representen fielmente esta información. Se puede manejar varios parámetros propios del audio para generar estos “ejes auxiliares”. A continuación, se destacan los siguientes:

Tabla 1.1: Parámetros de control en instrumentos digitales para síntesis aditiva de sonido [5].

Nivel	Descripción	Parámetro de control
Parcial	Señal Portadora (Seno)	<ul style="list-style-type: none"> • Tiempo inicio • Duración • Fase • Amplitud • Frecuencia
	Señales AM (tremolo) y FM (vibrato)	<ul style="list-style-type: none"> • Tipo de señal • Fase • Amplitud • Frecuencia
	Transitorios en amplitud y frecuencia	<ul style="list-style-type: none"> • Tamaño Máximo • Forma
Sonido	Timbre	<ul style="list-style-type: none"> • Relación del nivel parcial al sonido
	Volumen	<ul style="list-style-type: none"> • Tamaño Máximo • Forma
	<i>Glissando</i>	<ul style="list-style-type: none"> • Tamaño Máximo • Forma
	Crescendo y Decrescendo	<ul style="list-style-type: none"> • Tamaño Máximo • Forma
	Localización	<ul style="list-style-type: none"> • <i>Panning</i>
	Reverberación	<ul style="list-style-type: none"> • Duración • Tasa de descomposición

Según se puede observar en la Tabla 1.1, la cantidad de parámetros que pueden variarse para obtener una interpretación exitosa de datos es bastante variada. Sin embargo, el manejo de estos campos debe realizarse de manera particular para cada tipo de aplicación y teniendo en cuenta que un manejo irresponsable de los mismos puede llegar incluso a un deterioro de la salud del usuario.

Para estudiar la adaptación de datos hacía sonido, es importante entender ciertos parámetros inherentes al audio en sí, mismos que ya han sido estudiados en otras ramas, como la ingeniería de sonido y de manera general, la industria de la música.

1.4.1.1 Sonificación en Frecuencia y Amplitud con Señales Sinusoidales

Un método simple de sonificación, es interpolar los datos disponibles hacía señales periódicas simples, donde la frecuencia describa la relación de cambio entre los datos. La intercalación de frecuencias o tonos (*pitch*) es conocido comúnmente como melodía [6].

Una señal sinusoidal simple es perfectamente audible, siempre y cuando se manejen correctamente los valores de frecuencia (tono) y volumen (amplitud). La sinusoide cuenta con varios parámetros que pueden ser modificados para obtener resultados audibles distintos.

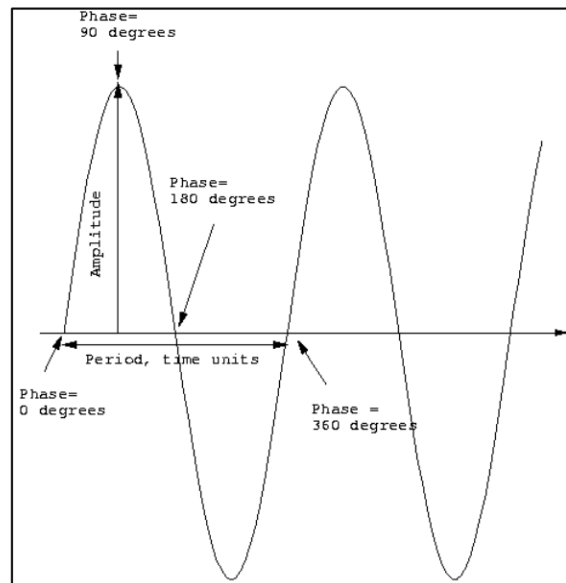


Figura 1.1: Interpretación gráfica de una sinusoide simple [36].

Los campos visualizables en la Figura 1.1, se explican de manera sencilla mediante la ecuación básica de una señal sinusoidal:

$$y(t) = A \sin (2\pi(ft + \emptyset)) \quad (1.1)$$

Donde:

- $t \rightarrow$ Define el tiempo.
- $A \rightarrow$ Define la amplitud de la sinusoide.

- $f \rightarrow$ Define la frecuencia de la senoide.
- $\emptyset \rightarrow$ Define la fase de la senoide.

En la Ecuación 1.1 es observable que existen diversas variables que pueden ser mapeadas con datos convencionales para obtener una retroalimentación auditiva. De esta forma una sonificación en amplitud y frecuencia, describe el cambio de los ejes referenciales base de la información hacia amplitud o frecuencia. El mapeo de datos puede realizarse de múltiples formas y existen otros parámetros que pueden modificarse para obtener una mejor interpretación auditiva.

1.4.1.2 Manejo de Timbre en Sonificación

El timbre es la cualidad del sonido que da matices a los distintos instrumentos musicales, es así, que dependiendo de la naturaleza propia de cada instrumento (cuerda, viento, materiales, etc.) se obtienen diferentes sonidos, aun cuando se toquen las mismas notas. A su vez, esta cualidad es aquella que observamos día con día, misma que brinda características propias a la voz de cada persona, el canto de un ave o el cualquier sonido natural [7]. La manera de modificar el timbre en señales generadas se produce mediante la adición de señales armónicas; estas se definen como señales cuya frecuencia es un múltiplo entero de una señal fundamental y que además su periodo coincide con el de la señal fundamental.

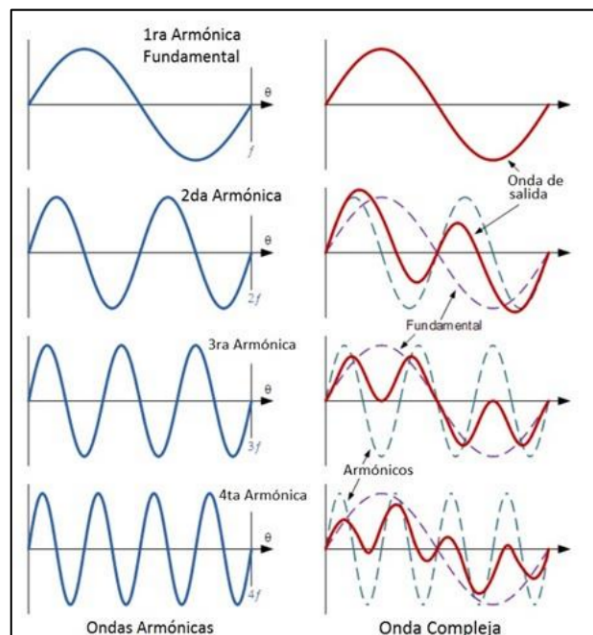


Figura 1.2: Comparativa inserción de armónicos [37].

En la Figura 1.2, se observa cómo la adición de una señal armónica otorga a la salida componentes no presentes en la señal fundamental original. La manera de encontrar cualquier armónico se puede observar a continuación:

$$h_n(t) = A \sin (2\pi(fnt + \phi)) \quad (1.2)$$

Donde:

- $n \rightarrow$ Describe el número de armónico a calcular.

Es así como, al sumar varios armónicos junto a una señal fundamental, se obtiene un resultado bastante diferente con respecto a una señal fundamental única, dotando de personalidad a un audio puntual, permitiendo que consumidores de este sonido obtengan un resultado más cercano a la realidad, mismo que permite una mejor interpretación de lo escuchado. Adicionalmente la amplitud (volumen) también será determinante acerca de en qué medida afecta el armónico a la señal final.

1.4.1.3 Especialización del Audio (Audio Espacial)

El audio espacial busca permitir al usuario obtener el sonido desde un origen físico especializado, de modo que sea posible identificar la posición del sonido en el espacio; técnica conocida como “*panning*”. La tecnología actual permite obtener esta experiencia, siempre y cuando los equipos de reproducción de sonido sean compatibles [8].

El Audio 5.1 busca una transmisión auditiva multicanal que logre la mayor inmersión dentro del consumo de recursos multimedia; es así como el nombre convencional de esta tecnología es “sonido envolvente” o *surround sound*.

La recomendación ITU-R BS.775-3 [9], define la utilización de 6 canales de audio; L (*Left*), R (*Right*), C (*Center*), LS (*Left Surround*), RS (*Right Surround*) y LFE (*Low Frequency Effects*). Se requerirá de al menos 4 bocinas para poder generar sonido envolvente, sin LFE³. La posición correcta de las bocinas es observable en la Figura 1.3.

Se puede estimar una posición geográfica mediante el uso correcto de los 5 canales definidos en ITU-R BS.775-3. Esto se producirá jugando con la transmisión de sonidos multicanal según sea necesario, además de tomar parámetros como el volumen (amplitud) del sonido. Los navegadores actuales son capaces de utilizar Audio 5.1, además la *Web*

³ LFE es un canal de audio opcional que funciona en una frecuencia inferior a los 120 [Hz].

Audio API es capaz de generar y reproducir sonido envolvente, aplicaciones de uso general como *YouTube* y *Netflix* utilizan esta tecnología siempre que esté disponible.

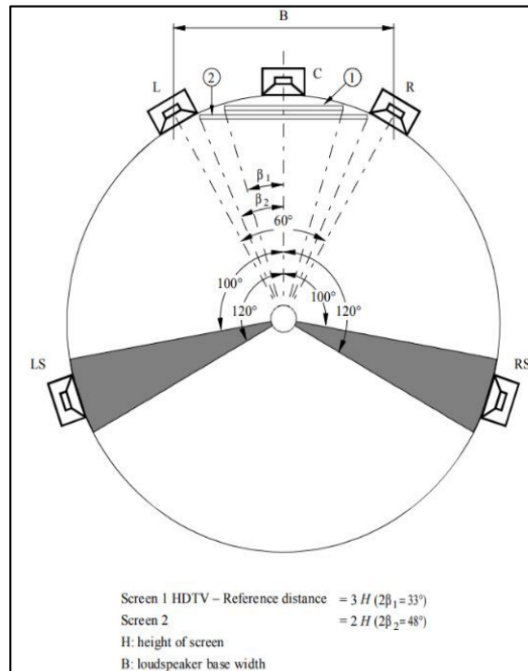


Figura 1.3: Esquema de posición de bocinas, ITU-R BS.775-3 [9].

1.4.2 HERRAMIENTAS Y TECNOLOGÍAS PARA LA IMPLEMENTACIÓN DEL PROTOTIPO

1.4.2.1 Google Spreadsheets

Spreadsheets es la herramienta desarrollada por *Google* para crear y modificar hojas de cálculo en línea, teniendo compatibilidad completa con *Microsoft Excel*. Para utilizar todas las funcionalidades de *Google Spreadsheets*, el único requerimiento es poseer una cuenta de Gmail, misma que ofrece acceso a todas las aplicaciones de *G-Suite* [10].

La fortaleza de *Spreadsheets* radica en su funcionamiento, puesto que no es necesario instalar nada (la aplicación funciona en una arquitectura web). Todo el procesamiento que realiza la hoja de cálculo se realiza en alguna instancia de *Google Cloud Platform*. A su vez es posible desarrollar *add-ons* o complementos para hojas de cálculo, de manera gratuita, mediante la plataforma *Apps Script*. Sin embargo, la publicación de los *add-ons* para ser distribuidos públicamente por *Google* requiere de un proceso de revisión y depuración, llevado a cabo por la propia empresa.

1.4.2.2 Google Apps Script

Google Apps Script es una plataforma que permite realizar tareas integradas a la mayoría de los productos de *Google (G-Suite)*. La plataforma busca simplificar el desarrollo en la misma utilizando una versión modificada de *javascript (googlescript)* para programación lógica y *HTML5* para la programación de interfaz de usuario [11].

El procesamiento de cualquier herramienta realizada sobre *Google Apps Script* se realizará de manera parcial entre un servidor propio de *Google*, mediante *Google Cloud Platform* y un navegador correspondiente al usuario que consume el aplicativo.

En el momento en que se crea un proyecto de *Apps Script* sobre una aplicación de *G-Suite*, se crea una instancia hacia *Google Cloud Platform*. Es así, que cualquier ejecución que requiera emplear *APIs* asociadas a la aplicación o manejar datos del usuario, se realizarán en la instancia de *Google Cloud* creada anteriormente [12].

Se puede asociar únicamente un proyecto de *Apps Script* por documento de *Google Spreadsheets* o cualquier otra aplicación de *G-Suite*; se pueden emplear varios *add-ons* a la vez, sin embargo, estos deben ser previamente verificados por la empresa y subidos a la sección de complementos.

Con respecto al desarrollo de *add-ons* sobre *Apps Script* para *Google Spreadsheets*, la sección lógica, deberá ser programada sobre el archivo *googlescript* de extensión “.*gs*”, sin embargo, es posible crear ficheros que contengan funciones de ejecución tradicionales en *javascript*, pero será necesario emplear una función sobre el documento *googlescript* para poder leer el contenido de estos ficheros. Toda la interfaz gráfica es programada sobre un fichero *HTML* que debe vincularse al archivo principal *googlescript*. Es posible importar librerías de estilo *CSS*, sin embargo, la documentación de *Apps Script* recomienda utilizar una librería propia al entorno [13].

La comunicación latente entre el servidor *Google Cloud* y el cliente en el navegador, puede afectar la comunicación del *add-on*, pues cada petición debe realizarse de manera completa en el servidor. Esto genera cierta latencia entre los procesos que varía dependiendo de la hora y el día [14].

1.4.2.3 Web Audio API

Esta API es una herramienta que permite manejar audio sobre la web, mediante *javascript*, generando búferes de audio que podrán ser creados, asociados o reproducidos según sea

necesario por la aplicación desarrollada [2]. Las herramientas de esta API permiten manejar el audio de una pestaña a diversos niveles, de modo que es posible modificar todo el audio de la pestaña a la vez.

a) Contextos de Audio

Los contextos de audio son entidades mediante los cuales es posible asociar elementos de audio a una pestaña puntual [15]. Para utilizarlos es necesario crear un nuevo contexto, posteriormente se crearán nodos de audio asociados al mismo.

```
1 var contexto_audio = new (window.AudioContext || window.webkitAudioContext)();
```

Código 1.1: Creación de un contexto de audio en *javascript*.

b) Nodo de Audio de Oscilador

Los osciladores son nodos de audio que esencialmente representan un generador de tonos de una onda pura, ya sea sinusoidal, diente de sierra o cuadrada [16]. La onda mencionada podrá ser modificada en amplitud, frecuencia y tiempo. Para crear un nodo de audio se debe asociar el nodo de oscilador a un contexto de audio previamente creado.

```
1 var oscilador_1 = contexto_audio.createOscillator();
```

Código 1.2: Creación de un oscilador asociado a contexto de audio en *javascript*.

c) Nodo de Audio de Fusión (Merger)

Los nodos de audio de tipo *merger* o fusión permiten tomar búferes de audio independientes y unirlos en un único búfer de audio [17], según se puede observar en la Figura 1.4. Este nodo es fundamental para manejar audio espacial en web, pues permite unir sonido en distintos canales en un único audio independiente.

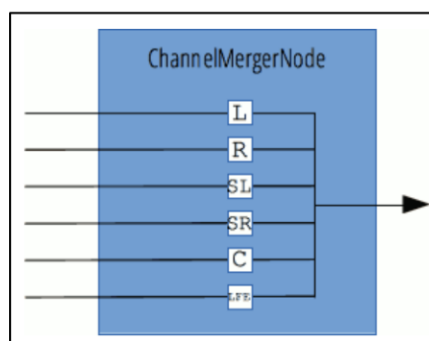


Figura 1.4: Diagrama de funcionamiento de un nodo de audio de fusión [17].

El nodo de fusión requerirá ser asociado a un contexto de audio y múltiples nodos de audio deberán conectarse a él para ser combinados. Adicionalmente, este nodo suele asociarse de manera directa a los parlantes a los que tiene acceso el navegador.

1.4.2.4 Web Speech API

La *API* de *Web Speech* es una herramienta que permite generar síntesis de voz basada en texto; funcionalidad comúnmente conocida como interpretación texto a voz o “*text to speech*”. Esta *API* está asociada internamente con la *Web Audio API*, por tanto, eliminar entidades como contextos de audio afecta al desarrollo de elementos de la *Web Speech API*. A su vez la programación del objeto debe realizarse sobre *javascript* [3].

Un elemento de *SpeechSynthesis* permite generar elementos de audio, que pueden leer texto y ofrecer una retroalimentación de lectura de este. Cualquier configuración sobre la interpretación texto a voz debe ser realizada sobre este elemento [18].

```
1 var synthesis = window.speechSynthesis;
```

Código 1.3: Definición de un elemento *SpeechSynthesis* sobre *javascript*.

1.4.3 METODOLOGÍA KANBAN

La metodología Kanban fue desarrollada por Toyota y esencialmente es una técnica que permite gestionar correctamente procesos de desarrollo; esto fundamentándose en la autogestión de los procesos de manera independiente [19].

Esencialmente se busca realizar exclusivamente los procesos de alto nivel que aseguren la realización de los procesos de menor nivel; de modo que se mantiene en desarrollo solo aquellos procesos que son fundamentales y contribuyen a la continuidad del proceso general. De esta manera se asegura la producción a tiempo (*just in time*).

La metodología Kanban consiste en generar un tablero en el cual se definirán de manera precisa las tareas que deben realizarse para completar un proyecto. El objetivo es separarlas en tres tipos de tareas: por completar, en progreso y completadas. De esta forma, es posible visualizar fácilmente qué tareas han sido completadas y cuáles otras requieren de atención y recursos para ser finalizadas. Las tareas colocadas en el tablero son en general simples, logrando que el flujo de trabajo sea continuo y siempre se esté trabajando en una tarea hasta finalizar completamente el proyecto.

2 METODOLOGÍA

En el presente capítulo se mostrarán los pormenores del *add-on* a realizar sobre la arquitectura de *Google Apps Script*, las decisiones de diseño tomadas y los distintos detalles de desarrollo realizados.

A lo largo del Apartado 2.1, Diseño, se definirán las principales pautas para el desarrollo del *add-on*; por otra parte, en el Apartado 2.2, Implementación, se explorarán las decisiones de desarrollo tomadas.

2.1 DISEÑO

2.1.1 PLANTEAMIENTO DE TABLERO KANBAN

Se utilizó la herramienta web *LucidSpark* [20] para realizar el tablero Kanban y actualizar los resultados a medida que se vayan consiguiendo los objetivos planteados.

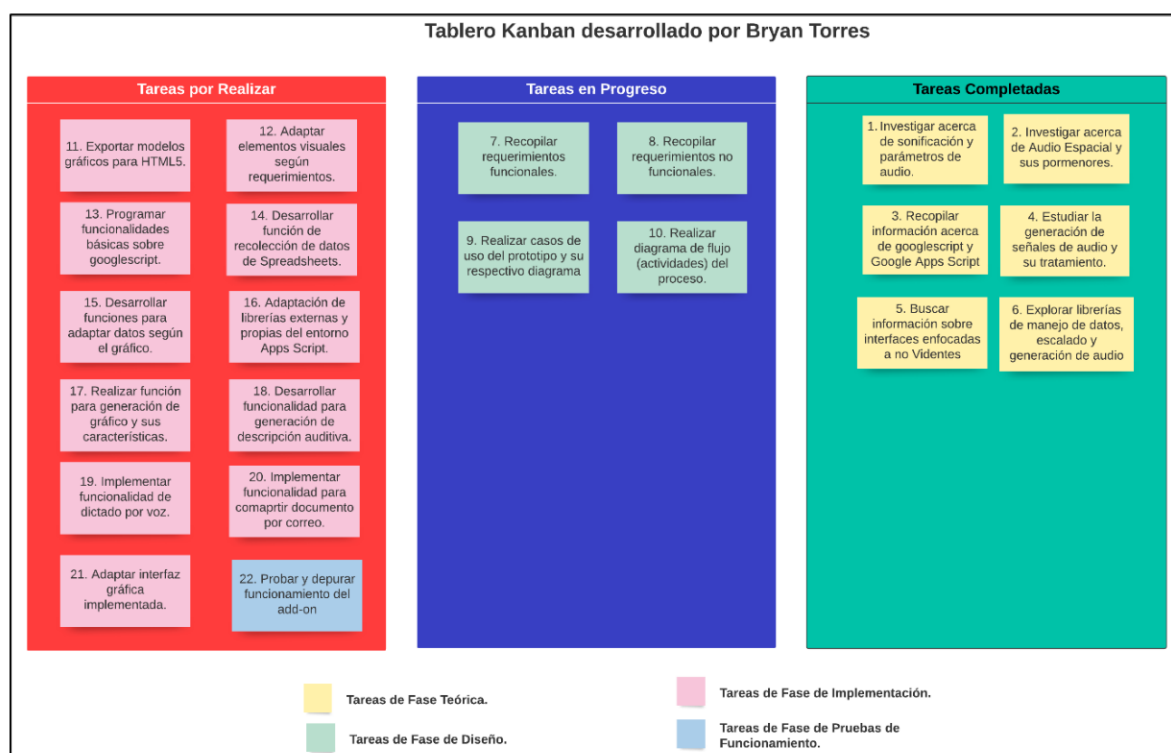


Figura 2.1: Planteamiento de Tablero Kanban en etapa de diseño.

Según se puede observar en la Figura 2.1, durante la etapa de diseño se deben realizar 4 tareas, cuya finalización correcta permitirá el desarrollo exitoso del *add-on* en cuestión.

2.1.2 DEFINICIÓN DE LA ARQUITECTURA DEL PROTOTIPO

Es complicado definir la arquitectura de una aplicación que funciona sobre la arquitectura de *Google Apps Script*, puesto que realmente ningún elemento, salvo los datos pertenecen al usuario. Esencialmente un *add-on* programado sobre *Google Apps Script* es un servicio web y tanto la lógica, como la presentación y los datos residen en los servidores de *Google* mediante *Google Cloud Platform*.

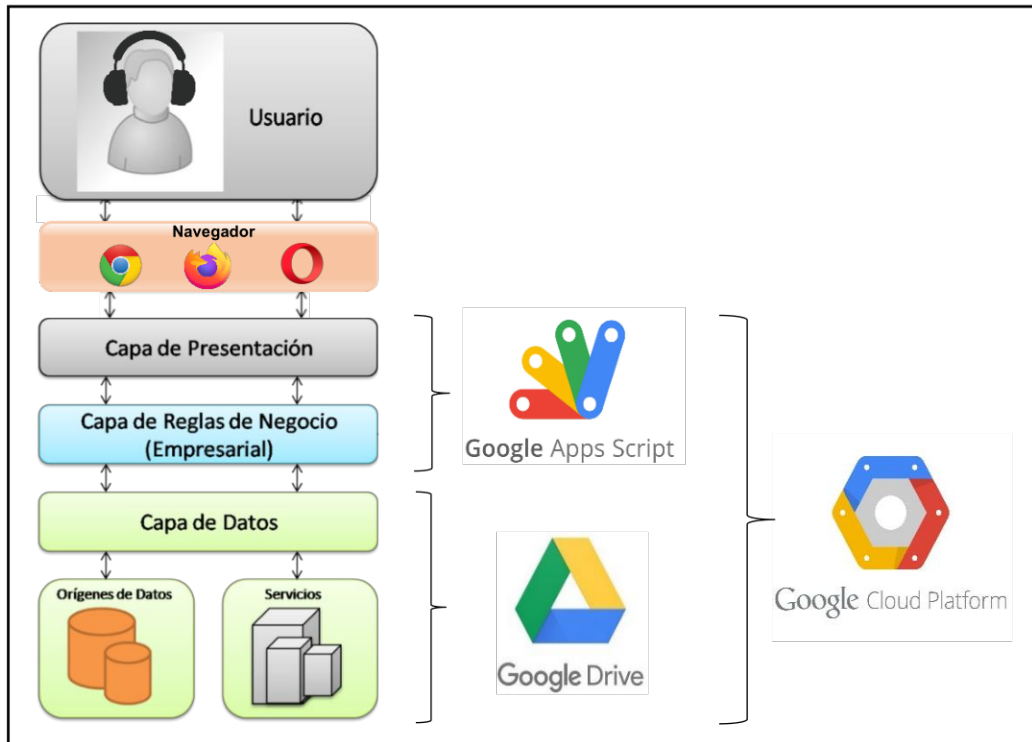


Figura 2.2: Arquitectura de desarrollo por capas para *add-on* [38] [39] [13].

En la Figura 2.2 se puede observar una arquitectura de desarrollo por capas tradicional, es así como las capas de Presentación y Negocio sobre el *add-on* serán gestionados valiéndose de *Apps Script*; adicionalmente, el uso de *APIs* y servicios, ya sean de *Google* o no deberá ser también definido durante el desarrollo de estas dos capas. Por otra parte, los datos de los usuarios (aquellos contenidos en la hoja de cálculo) serán almacenados y gestionados mediante *Google Drive*. Todo el ecosistema formado por estas tecnologías será administrado a su vez por una instancia en *Google Cloud Platform*, instancia que se encuentra ejecutándose sobre *Google Spreadsheets* [13].

Si bien la instancia de *Google Cloud Platform* siempre está ahí, no es visualizable, esta es transparente al usuario y se torna visible en el momento en que se crea el proyecto de *Apps Script*.

Es así como las capas a desarrollarse corresponderían a las de Presentación y Negocio, pero a su vez se destaca que debido a las características propias del entorno el límite entre una y otra es difuso.

2.1.3 REQUERIMIENTOS FUNCIONALES Y NO FUNCIONALES DEL PROTOTIPO

La puntualización de requerimientos tanto funcionales como no funcionales, es una etapa fundamental para el correcto desarrollo de cualquier prototipo. Durante esta sección se definirán todas las características que deben cumplirse en la implementación final.

Previo a la definición de requerimientos es importante comprender que las personas con discapacidad visual, en todos los niveles utilizan usualmente tecnología y se valen principalmente de dos tipos de dispositivos complementarios: lectores de pantalla y pantallas de tipo braille; sin embargo, esta última puede llegar a tener precios exageradamente altos además de ser relativamente lentas por la falta de desarrollo del campo [21]. De esta forma, usuarios con discapacidad visual utilizan ordenadores tradicionales con la ayuda de lectores de pantalla.

Adicionalmente, se debe destacar que, dentro del propio segmento de personas con discapacidad visual, existirán aquellos que hayan nacido con esta eventualidad o personas que debido a enfermedades u otras causales accidentales hayan adquirido esta característica con el tiempo. Es así, que el desarrollo de aplicaciones de cualquier índole enfocado en este segmento de la población debe necesariamente centrarse a aquellos que hayan nacido con esta condición, puesto que el prototipo desarrollado debe poder ser manejado con una facilidad relativamente similar por los dos tipos de usuarios no videntes definidos previamente [21].

2.1.3.1 Requerimientos Funcionales (RF)

Los requerimientos funcionales definen aquellas funcionalidades que deben necesariamente incluirse en la versión final del *add-on* con el fin de cumplir los objetivos planteados en un inicio, tanto como aquellos que van surgiendo durante el desarrollo del proyecto.

Debido a las cualidades del público hacia el que va orientado el *add-on*, existirán requerimientos funcionales específicos para un segmento de la población y otros que son generalidades de diseño de software.

Dichos requerimientos son planteados a continuación:

- RF1.** Necesariamente se debe dar una retroalimentación auditiva a lo largo de todo el flujo de acción del prototipo, de modo que se debe dotar al *add-on* de un método de síntesis y reproducción de voz (*text to speech*) de modo que, durante las diversas etapas del software, el usuario esté totalmente consciente de lo que está sucediendo y de qué acciones debe realizar para obtener el resultado esperado. Se plantea el uso de síntesis por voz pues es uno de los métodos más empleados por usuarios no videntes [21] para utilizar tecnologías de ofimática y similares.
- RF2.** Será necesario añadir eventos del tipo de retroalimentación negativa (error) o positiva (correcto), puesto que dependiendo del evento puede ser poco eficiente el uso de una descripción hablada mediante síntesis de voz.
- RF3.** Se buscará que la forma de ingreso de datos sea el mismo manejado por *Google Spreadsheets*, puesto que ya existen maneras de gestionar la inserción de datos en hojas de cálculo enfocadas a usuarios no videntes (principalmente con lectores de pantalla) [22] y es posible que una buena parte del público objetivo ya utilice estos métodos.
- RF4.** Los datos encontrados en la hoja de cálculo deberán ser seleccionables mediante distintos métodos para su posterior sonificación. Se buscará emplear métodos que impliquen tanto el ratón como el teclado. La diversidad de métodos permite a usuarios no videntes, como videntes, una fácil adaptación a cualquier herramienta de software.
- RF5.** Los usuarios deben de tener una manera sencilla de seleccionar el tipo de gráfico que desean sonificar, permitiéndole elegir entre los cuatro tipos de gráficos planteados: gráficos de línea, de barras, de pastel y de dispersión. Se debe priorizar el uso del teclado, proveyendo adicionalmente maneras alternativas de seleccionar esta opción (Es probable que algunos usuarios usen la herramienta con la ayuda de otros individuos que no necesariamente cuenten con algún tipo de discapacidad visual).

- RF6.** Se debe dotar al usuario de métodos eficientes para lidiar con valores faltantes o inconsistentes. Este tipo de datos pueden sesgar los resultados finales; además, están presentes en la gran mayoría de fuentes de información [23]. Sin embargo, el usuario debe ser quién decida qué hacer con dichos datos.
- RF7.** El usuario debe de tener una forma eficiente de compartir la hoja de cálculo editada con otros usuarios, tengan o no algún tipo de discapacidad visual. Se debe proveer una manera sencilla de compartir el documento modificado mediante correo electrónico. Esto además permitirá usar la funcionalidad de *Google Spreadsheets*, de edición en tiempo real por varios usuarios del documento.

2.1.3.2 Requerimientos no Funcionales (RNF)

Estos requerimientos definen parámetros complementarios a los definidos por aquellos funcionales. Mediante ellos se dota de calidad al software final y a su vez se asegura la consecución de los objetivos planteados inicialmente. Dichos requerimientos son planteados a continuación:

- RNF1.** La interfaz visible debe ser simple y ordenada, pues es probable que usuarios no videntes se valgan de ayudantes para utilizar el prototipo. Se priorizará el uso de los elementos gráficos definidos por *Google* para desarrolladores de *Apps Script*.
- RNF2.** La interfaz debe de ser eficiente, de modo que los tiempos de procesamiento sean principalmente derivados de la comunicación entre el cliente y *Google Cloud*.
- RNF3.** El flujo de ejecución debe de ser lógico y ordenado, de modo que se disminuya la posibilidad de cometer errores por parte del usuario en el ingreso de datos o parámetros propios del gráfico. Existirán controles que eviten el desbordamiento de parámetros o la inconsistencia de la sonificación.
- RNF4.** La cantidad de usuarios capaces de utilizar una misma hoja de cálculo con el *add-on* a la vez, será la definida por *Google Spreadsheets*; sin embargo, se asegura el uso de no más de 5 personas simultáneamente. Se destaca que las sonificaciones generadas serán solo audibles por la persona que así lo

requiera, las demás personas serán capaces de ver los datos utilizados para la sonificación y el gráfico por el mismo dentro de la hoja de cálculo.

- RNF5.** El *add-on* deberá ser compatible con al menos tres navegadores. Es importante garantizar que el software desarrollado tenga la máxima compatibilidad, de modo que se puedan aprovechar las funcionalidades extras proporcionadas por las aplicaciones del entorno *G-Suite*. Se probará su uso en los navegadores *Google Chrome, Opera y Edge* (su uso será delimitado a equipos de escritorio, pues adaptar una interfaz del estilo a equipos móviles escapa del alcance de este Trabajo de Integración Curricular).
- RNF6.** No existirán demasiados eventos auditivos al mismo tiempo, de modo que no se sature al usuario con diversa información al mismo tiempo. Esto se conseguirá mediante el control ordenado del flujo de ejecución. Adicionalmente, los sonidos generados deberán tener un respectivo espaciamiento temporal para evitar superponer contenido.
- RNF7.** El *add-on* no deberá tener demasiadas opciones, de modo que su uso sea sencillo e intuitivo; los atajos por teclado implementados deben buscar la reusabilidad en caso de ser posible.
- RNF8.** Los permisos solicitados al usuario serán exclusivos del uso correcto del *add-on*. No se utilizará información ajena al mismo, ni se compartirá o almacenará información personal de ningún tipo.
- RNF9.** El tamaño de visualización del *add-on* no deberá ser superior al 30% del tamaño total del entorno de *Google Spreadsheets* o inferior al 5% del tamaño total definido, esto con el fin de evitar entorpecer el uso de la hoja de cálculo, pero a su vez evitando que sea complicado encontrarlo dentro de la pantalla.
- RNF10.** El *add-on* deberá ser posible de ejecutar dentro de las funciones propias de *Google Spreadsheets*. Evitar que el usuario tenga que acceder a sitios externos favorecerá su usabilidad.
- RNF11.** El sonido generado debe tener un rango audible adecuado, de modo que no pueda provocar perjuicios a la salud de los usuarios.

2.1.4 DIAGRAMA DE CASOS DE USO DEL PROTOTIPO

El diagrama de Casos de Uso permite mostrar el comportamiento base del sistema desde el punto de vista de un actor o usuario. De esta manera, se establecen las relaciones y roles entre el usuario y el sistema en sí. Se utilizó la herramienta en línea *LucidChart* [24] para realizar el diagrama de casos de uso. El mencionado diagrama correspondiente al *add-on* desarrollado se puede observar a continuación:

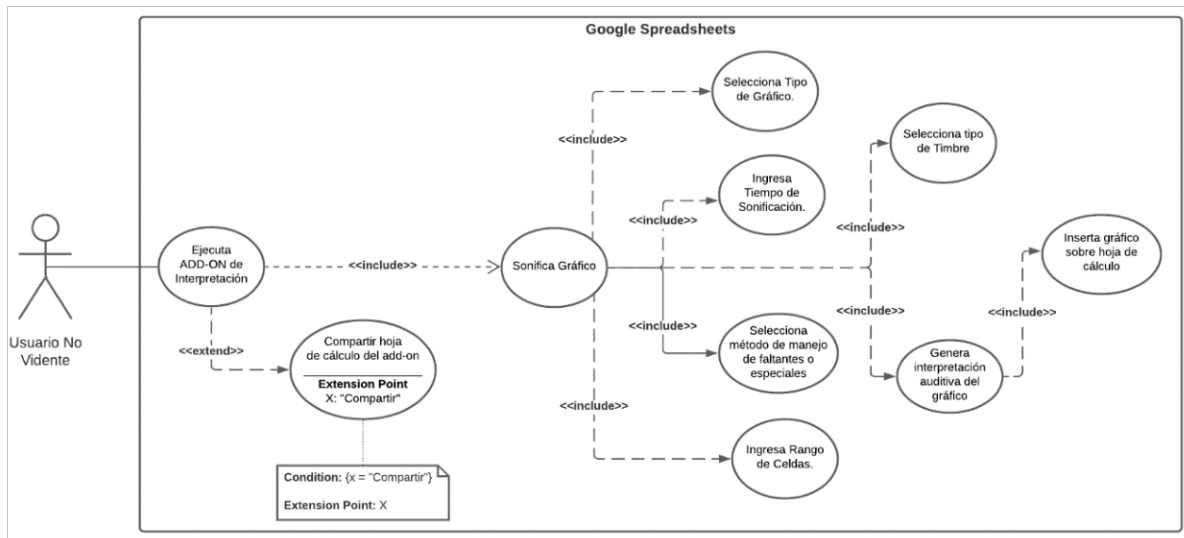


Figura 2.3: Diagrama de Casos de Uso del prototipo.

En el diagrama de Casos de Uso, observable en la Figura 2.3 se definen como actor al usuario no vidente, individuo que contará al que se dotará de funcionalidades suficientes para realizar una sonificación eficiente. Las relaciones de inclusión permiten establecer todas las funciones disponibles del *add-on*, estas se derivan de la ejecución del mismo.

Adicionalmente, se usa una relación de extensión para la compartición de la hoja de cálculo debido a que esta funcionalidad será opcional y requerirá que el usuario solicite de manera explícita que desea compartir el documento *Spreadsheets* de hoja de cálculo.

Se destaca que el diagrama de casos de uso contempla el uso del prototipo una vez el usuario tiene acceso a la plataforma de *Google Spreadsheets*. Previamente se debe ingresar con una cuenta de *Gmail* y crear un archivo de *Spreadsheets*, sin embargo, estos pasos no son definidos pues escapan del alcance de este Trabajo de Integración Curricular.

2.1.5 DIAGRAMA DE ACTIVIDADES DEL PROTOTIPO

Mediante el Diagrama de Actividades podemos modelar la secuencia lógica de funcionamiento de un sistema. De este modo se puede tener en claro el orden de ejecución de secciones, las condiciones previas a su ejecución, además de permitir definir variables que serán necesarias para el correcto desenvolvimiento del prototipo. A continuación, se observa el Diagrama de Actividades realizado sobre *LucidChart*:

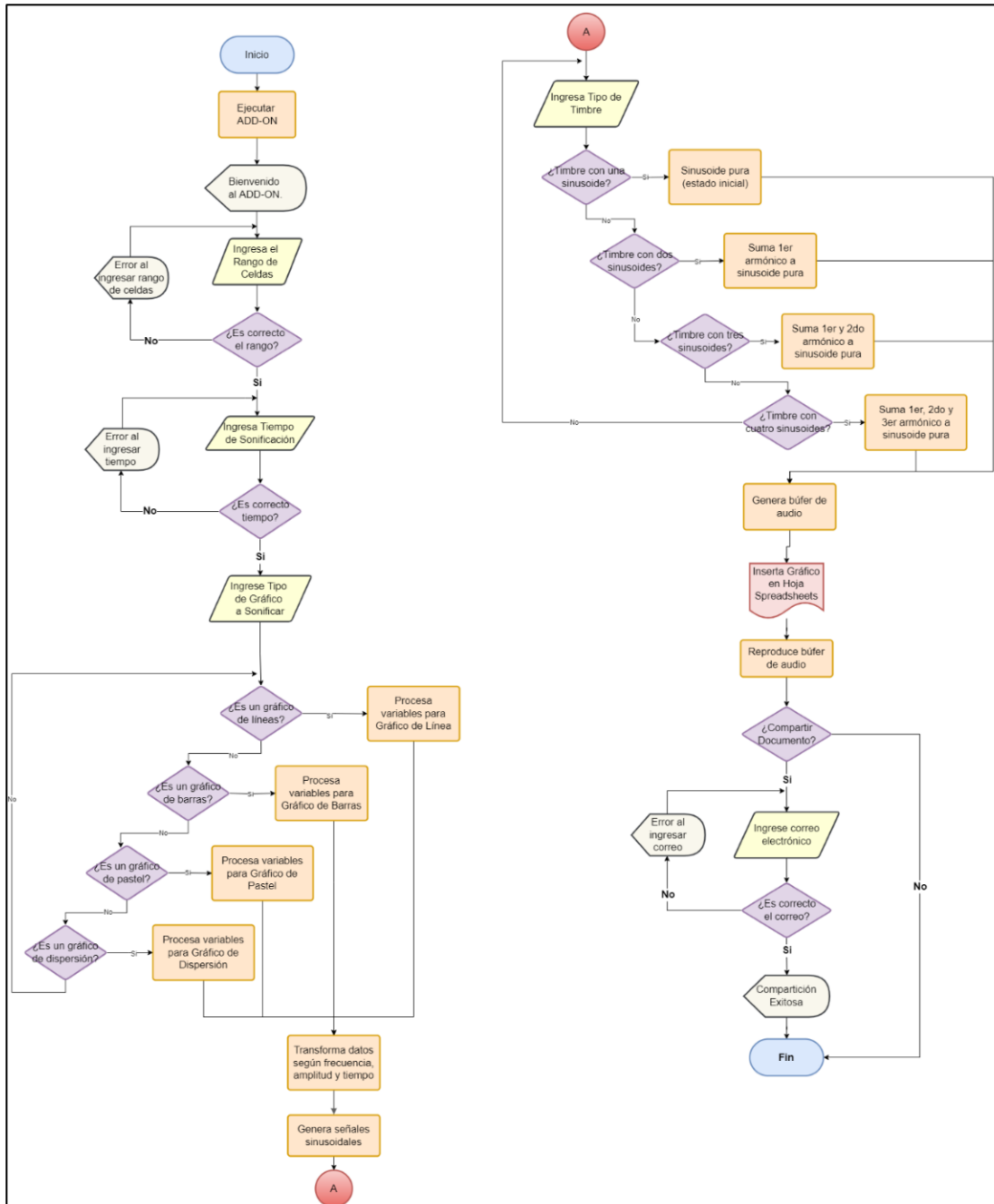


Figura 2.4: Diagrama de Actividades del prototipo.

El Diagrama de Actividades observable en la Figura 2.4, define mediante palabras varios procesos. Si bien es posible definir el flujo mediante el uso de variables y operaciones matemáticas, su interpretación se tornaría bastante más complicada, perdiéndose completamente el objetivo del gráfico.

2.1.6 DISEÑO DE LA INTERFAZ

En el prototipo desarrollado, la interfaz gráfica jugará un papel más bien secundario. No obstante, la interfaz de manera general en el mundo del software posee una definición mucho más abierta; esta se define como aquel elemento que permite la correcta interacción entre máquina y usuario, otorgando una retroalimentación continua a lo largo de un determinado proceso [25].

2.1.6.1 Diseño de Interfaz Gráfica

Si bien el *add-on* está enfocado a un público con discapacidad visual es probable que este se use con la ayuda de un individuo de soporte (con el fin de agilizar el proceso) o por otra parte, que el mismo quiera utilizarlo empleando nada más la ayuda de un lector de pantalla. En cualquiera de los dos casos, es necesario que una interfaz gráfica sirva como medio de interconexión entre el usuario y la lógica detrás del prototipo.

Los sitios web actuales adoptan varias de las tecnologías propuestas por la WCAG⁴ [1], de modo que la mayoría de los navegadores son compatibles con lectores de pantalla, ya sea de terceros o embebidos dentro del propio software. Sin embargo, los lectores de pantalla cuentan con un funcionamiento lineal, describiendo todo por cuanto el cursor se pose, salvo elementos definidos como no legibles. Por esta razón el descubrimiento de páginas web es un reto que toma bastante tiempo a usuarios con discapacidad visual. Por dicho motivo, se plantean ciertos lineamientos para el desarrollo de interfaces gráficas enfocadas en usuarios no videntes [26]:

- Desarrollar interfaces gráficas cuyo flujo de trabajo pueda ser adaptable a una ejecución secundaria empleando retroalimentaciones auditivas y comandos de voz o teclado.
- Codificar interfaces limpias, es decir, que no posean demasiados elementos juntos o que sobresaturen de texto varias secciones. Debido al funcionamiento de lectores

⁴ WCAG (Web Content Accessibility Guidelines) corresponde a una serie de recomendaciones que buscan brindar acceso a la web a usuarios con diversas discapacidades.

de pantalla secciones con demasiados elementos pueden ser complejos de interpretar.

- Aplicar la guía ETI (*Enhanced Text Interface*), para ubicación de elementos visuales.

Tabla 2.1: Guía ETI (interfaz de texto mejorada) para desarrollo web [26].

Nº	Recomendación	Base de Desarrollo
1	Comunicar la estructura del menú a través de numeración o etiquetas.	Memorizar posiciones fácilmente.
2	Etiquetar todos los elementos gráficos empleados.	Generar modelos mentales.
3	Colocar botones en línea o después de opciones o cuadros de texto.	Demostración mediante estudio.
4	No utilizar descripciones extensas.	Memorizar posiciones Fácilmente
5	Utilizar elementos de navegación genérica, con pocos elementos visuales.	Modelos mentales, consistencia web.
6	Colocar links externos en la parte inferior y barras de búsqueda en la parte superior.	Modelos mentales, consistencia web.
7	Colocar elementos sin funcionalidad aparente en la parte más baja de la página.	Modelos mentales, consistencia web.
8	Evitar saturar con elementos visuales de marca, de existir colocarlos al inicio al final.	Consistencia web, demostración estudio.

- Utilizar la etiqueta “*alt*”, referente a “*alternative text*”, dentro de elementos HTML. La etiqueta “*alt*” funciona de manera similar a las leyendas en gráficos tradicionales, permitiendo leer por medio de lectores de pantalla la descripción colocada. Esta funcionalidad puede ser empleada en elementos diversos como imágenes (<*img*>), divisores (<*div*>), botones (<*button*>) o incluso elementos de párrafo (<*p*>) [27].



Figura 2.5: Imagen que poseía una etiqueta *alternative text*, interpretada mediante lenguaje *HTML*.

El diseño de la interfaz gráfica desarrollada de manera final seguirá todos los parámetros anteriormente descritos, de modo que se facilite la integración con personas no videntes.

2.1.6.2 Diseño de Interfaz Auditiva

En contra parte, con lo anteriormente planteado, pero retomando el primer Apartado planteado en las puntualizaciones realizadas; lo más importante a la hora de desarrollar una interfaz, con utilidad real para usuarios no videntes, es entender que se deben desarrollar dos interfaces: una interfaz sin enfoque gráfico que será desarrollada exclusivamente para un público con discapacidad visual y una interfaz gráfica cuyo flujo de trabajo es complementario. Con esto en mente, el desarrollo de la interfaz auditiva seguirá los siguientes preceptos:

- Evitar que varios sintetizadores de voz suenen a la vez o utilicen demasiadas palabras (procurar el uso de palabras clave). Varios lectores leyendo demasiadas palabras al mismo tiempo o diciendo demasiadas palabras suele confundir al usuario final, en lugar de ofrecer una ayuda para la interpretación.
- Utilizar atajos de teclado complementariamente al cursor, o reemplazando totalmente su funcionalidad. Los atajos de teclado pueden facilitar el uso de interfaces de usuario, dado a que se puede mapear funciones a teclas específicas, ofreciendo experiencias rápidas y con menor cantidad de errores [28].
- Utilizar atajos de teclado conocidos, combinando teclas de acción comunes como SHIFT o ALT sumado a alguna otra tecla. De esta manera, se evitará interferir con comandos propios del sistema o de la aplicación sobre la que se esté desarrollando.
- Utilizar teclas que sean fácilmente reconocibles, como aquellas con formas diferentes o con patrones distintivos [29]. Los teclados regulares cuentan con protuberancias en teclas como la letra “F” y “J”, estos identificadores permiten facilitar la escritura en ordenadores sin mirar constantemente el teclado, esta costumbre se trae desde ciertas prácticas de mecanografía anteriores a los ordenadores.
- Desarrollar el prototipo siguiendo un flujo lógico, mismo que sea lineal y evite proveer demasiadas opciones al usuario. De modo que las opciones se muestren una tras otra, buscando reducir la generación de errores por inserción errónea o por saltarse pasos necesarios para el funcionamiento correcto del sistema.

- Proveer, siempre que sea posible y lógico retroalimentaciones auditivas, como: sonidos de alarma, voces humanas pregrabadas o de sintetizadores de voz. Además de acompañar elementos visuales de etiquetas de texto alternativo, para facilitar la integración con interfaces lectores de pantalla.

Las puntualizaciones mencionadas tanto en los Apartados 2.6.1.1 y 2.6.1.2, se utilizarán en conjunto, con el objetivo de proveer una interfaz de usuario sencilla, que evite el desbordamiento de información hacia el usuario y a su vez prevenga en lo posible la generación de errores por malentendidos en el uso del prototipo.

2.2 IMPLEMENTACIÓN

Durante esta sección se presentará todo el proceso de desarrollo y codificación que lleva a la consecución del *add-on* de interpretación gráfica sobre *Google Spreadsheets*.

2.2.1 ACTUALIZACIÓN DEL TABLERO KANBAN

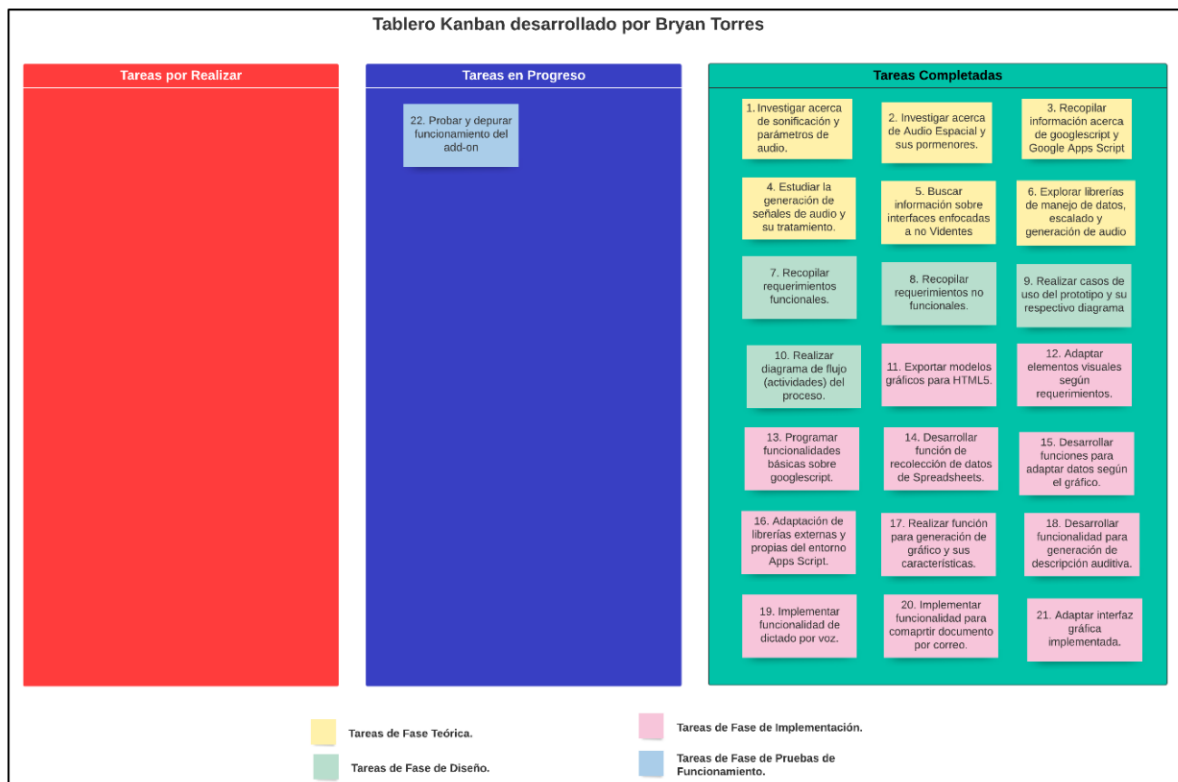


Figura 2.6: Actualización del tablero Kanban en etapa de Implementación.

En la Figura 2.6 en la columna de “Tareas Completadas” se observan aquellas tareas ya finalizadas, además de visualizar aquellas tareas a realizarse a lo largo de esta etapa.

Se destaca que la cantidad de tareas es superior con respecto a las etapas anteriores, sin embargo, las tareas relacionadas con la etapa de implementación son más bien modulares.

2.2.2 CODIFICACIÓN DEL DIAGRAMA DE ACTIVIDADES

En la Figura 2.4, es observable que el *add-on* sigue una estructura lógica bien establecida de modo que se seguirá un orden concreto para el desarrollo sus distintas funcionalidades. Las etapas realizadas durante el presente apartado buscan satisfacer los parámetros planteados a lo largo de la Etapa de Diseño.

2.2.2.1 Punto de Ejecución del *Add-on*

Un punto de ejecución describe las condiciones necesarias para que el *add-on* pueda ser ejecutado por un usuario cualquiera. Para que el *add-on* pueda ser ejecutable desde la interfaz gráfica de *Google Spreadsheets*, es necesaria la realización de funciones que permitan insertar funcionalidades adicionales dentro del menú propio de la plataforma y que ayuden a crear la instancia de *Google Cloud Platform*. Para cumplir con este objetivo se crearon tres funciones específicas definidas en la documentación de *Apps Script*:

- a) Función “onOpen”: esta función determina la secuencia de comandos que deben agregarse al *add-on* para poder iniciarse sobre *Google Spreadsheets*.

```
1 function onOpen(e) {
2   SpreadsheetApp.getUi().createAddonMenu()
3     .addItem('Ejecutar Add-On', 'mostrarComplemento')
4     .addToUi();
5   mostrarComplemento()
6 }
```

Código 2.1: Creación punto de ejecución, desde *googlescript*.

En el Código 2.1, se crea un nuevo menú dentro del objeto *SpreadsheetApp*, correspondiente a la interfaz de la hoja de cálculo, permitiendo ejecutar una función que permita mostrar dentro de la interfaz al *add-on*. La función “mostrarComplemento” apuntará a un archivo *HTML*, permitiendo que se muestre en pantalla la interfaz del *add-on*, en conjunto con la propia de *Apps Script*.

```
1 function mostrarComplemento() {
2   var ui = HtmlService.createTemplateFromFile('barra_add_on')
3     .evaluate()
4     .setTitle('Add-On de interpretación gráfica')
5     .setSandboxMode(HtmlService.SandboxMode.IFRAME);
6
7   SpreadsheetApp.getUi().showSidebar(ui);
8 }
```

Código 2.2: Creación función de asociación de interfaz gráfica, en *googlescript*.

- b) Función “onInstall”: esta función permite definir que funciones se agregaran de manera predeterminada a la instancia de *Apps Script*.

```
function onInstall(e) {
  onOpen(e);
}
```

Código 2.3: Definición función a ejecutarse durante ejecución, en *googlescript*.

- c) Función “doGet”: esta función permite definir un archivo de extensión *HTML*, para apuntar al mismo, permitiendo pasar contenido entre el cliente en *Spreadsheet* y las funciones del servidor en *Apps Script* o en archivos independientes *javascript*.

```
1 function doGet() {
2   var template = HtmlService.createTemplateFromFile('barra_add_on');
3   return template.evaluate().setSandboxMode(HtmlService.SandboxMode.IFRAME);
4 }
```

Código 2.4: Función de paso de contenido entre *Google Spreadsheets* y cliente *HTML*.

2.2.2.2 Captura de Datos Desde Spreadsheets

Al momento de intentar capturar datos desde la hoja de datos hacia *Apps Script* es importante comprender que el *add-on* es una vista *HTML* distinta a la interfaz de *Google Spreadsheets* y la comunicación debe realizarse llamando siempre a la instancia *SpreadshetApp*. Los datos necesarios para el proceso de sonificación serán extraídos en dos pasos:

- a) Capturar Rangos de Datos: para capturar el rango de datos se plantean dos opciones, capturar el rango de datos seleccionados en *Google Spreadsheets* o tomarlos como un rango desde un cuadro de texto en la vista *HTML* del *add-on*.

```
1 function extraer_valores(){
2   var mi_app = SpreadsheetApp;
3   var hoja_actual = mi_app.getActiveSpreadsheet().getActiveSheet();
4   var celdas = hoja_actual.getActiveRange().getValues();
5
6   return celdas
7 }
```

Código 2.5: Función para extraer rango activo de celdas, en *googlescript*.

Durante los pasos anteriores, se utilizaron funciones programadas en *googlescript*, sin embargo, será necesario realizar ciertas funciones desde eventos lógicos adicionales sobre *javascript*. Para pasar un dato desde la vista *HTML* hacia *Apps*

Script requiere de cargar el valor de una variable y correr el script desde el lado del cliente.

```
1 let prueba = google.script.run.withSuccessHandler(  
2   function (fuente)  
3   {  
4     let prueba = JSON.parse(fuente);  
5  
6     console.log("El rango seleccionado es: " + JSON.parse(fuente));  
7   }).verificar_rango(document.getElementById("rango").value);
```

Código 2.6: Ejecución de función *googlescript* desde *javascript*.

El Código 2.6 muestra que los datos de la vista *HTML* y *Apps Script* no están en una misma capa, por tanto, deben comunicarse realizando una transformación *JSON*. Adicionalmente para llamar la función se debe utilizar la función específica "*google.script.run*", la cual permite correr una función de *Apps Script*, desde el lado del cliente.

En la línea 1, se puede observar que se debe definir un manejador de exitoso o fracaso, de modo que el Cliente tenga programada una funcionalidad tanto en caso de que la ejecución falle o se complete correctamente. Las líneas desde la 2 hasta la 5 describen el comportamiento a realizar en caso de que se ejecute correctamente la función "verificar_rango".

```
1 function verificar_rango(rangazo){  
2   let mi_app = SpreadsheetApp;  
3   let hoja_actual = mi_app.getActiveSpreadsheet().getActiveSheet();  
4   let celdas = [];  
5   if(!rangazo.match(/S/)){  
6     try{  
7       celdas = hoja_actual.getActiveRange().getValues();  
8       return JSON.stringify("CORRECTO");  
9     }  
10    catch(err){  
11      return JSON.stringify("ERROR_SELECT");  
12    }  
13  } else{  
14    try{  
15      let rango = hoja_actual.getRange(rangazo);  
16      rango.activate();  
17      celdas = hoja_actual.getRange(rangazo).getValues();  
18      return JSON.stringify("CORRECTO");  
19    }  
20    catch(err){  
21      return JSON.stringify("ERROR_INSERT");  
22    }  
23  }
```

Código 2.7: Función de carga de datos a ser llamada en la línea 7 del Código 2.6 , desde *googlescript*.

Los datos serán extraídos desde las celdas de la propia hoja de cálculo y se transformarán en matrices. Las matrices describen esencialmente tablas que pueden ser fácilmente manipuladas tanto desde *javascript* como *googlescript*.

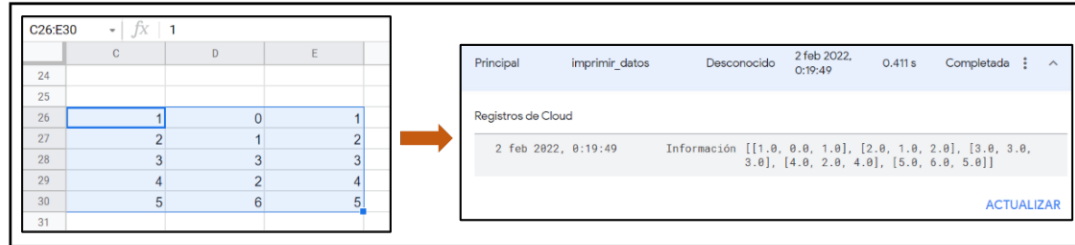


Figura 2.7: Pase de datos en celdas de *Google Spreadsheets* a matriz *javascript*, observado desde consola de *Apps Script*.

Según se observa en la Figura 2.7, una matriz se define de la misma manera que un vector. Por tanto, las matrices son arreglos ordenados de datos. Es así como las funciones definidas para vectores son igualmente utilizables con matrices, facilitando el proceso de implementación.

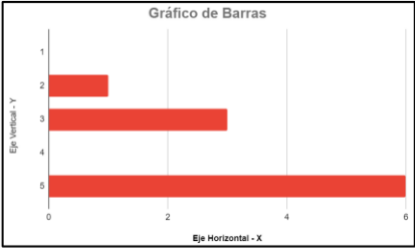

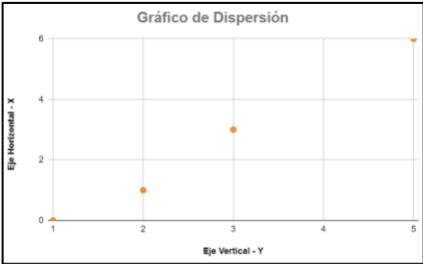
- b) Extracción del tiempo de sonificación: el usuario no vidente insertará este valor y este posteriormente será pasado a *Apps Script*. El tiempo se proveerá a la interfaz como un cuadro de texto, de modo que el usuario sea capaz de insertarlo desde la interfaz gráfica. El método para devolver el valor de la variable de tiempo es el mismo utilizado en el Código 2.6, línea 7.

2.2.2.3 Definición de Ejes Auditivos Complementarios

Los gráficos tradicionales cuentan con ejes de coordenadas que sirven de referencia para entender el gráfico, comúnmente (x, y, z). Los gráficos definidos cuentan con un máximo de dos ejes y un mínimo de uno, es decir se trata de gráficos unidimensionales y bidimensionales. La definición de los nuevos ejes es descrita a continuación:

Tabla 2.2: Definición de ejes complementarios y características adicionales.

Tipo de Gráfico	Ejes y Características
<p>Gráfico de Línea:</p>	<p>Ejes complementarios:</p> <ul style="list-style-type: none"> • Eje X → Espacial, canales de audio estéreo. • Eje Y → Frecuencia, valor máximo asociado a frecuencia más alta. <p>Características adicionales:</p> <ul style="list-style-type: none"> • Amplitud de la senoide: [0.2 – 1] para cada señal pura.

<p>Ejes:</p> <ul style="list-style-type: none"> • Tipo → Bidimensional. • Eje X → Horizontal. • Eje Y → Vertical. 	<ul style="list-style-type: none"> • Tiempo: tiempo de sonificación escalado según distancia. <p>Características sinusoides armónicas:</p> <ul style="list-style-type: none"> • Amplitud: 0.8 para cada armónico. • Tiempo: equivalente a la señal original.
<p>Gráfico de Barras:</p>  <p>Ejes:</p> <ul style="list-style-type: none"> • Tipo → Bidimensional. • Eje X → Horizontal. • Eje Y → Vertical. 	<p>Ejes complementarios:</p> <ul style="list-style-type: none"> • Eje X → Espacial, canales de audio estéreo, además de duración del tiempo. • Eje Y → Frecuencia, cada barra escalada a una nota musical. <p>Características adicionales:</p> <ul style="list-style-type: none"> • Amplitud de la sinusoide: [0.2 – 1] para cada señal pura. • Tiempo: tiempo de sonificación escalado a mayor barra, las demás con el valor equivalente. <p>Características sinusoides armónicas:</p> <ul style="list-style-type: none"> • Amplitud: 0.8 para cada armónico. • Tiempo: equivalente a la señal original.
<p>Gráfico de Pastel:</p>  <p>Ejes:</p> <ul style="list-style-type: none"> • Tipo → Unidimensional. • Eje → Expresa porcentaje. 	<p>Ejes complementarios:</p> <ul style="list-style-type: none"> • Eje → Frecuencia, cada pedazo escalado a una nota musical y magnitud representada mediante tiempo. <p>Características adicionales:</p> <ul style="list-style-type: none"> • Amplitud de la sinusoide: [0.2 – 1] para cada señal pura. • Tiempo: tiempo de sonificación multiplicado por el porcentaje correspondiente del pedazo. <p>Características sinusoides armónicas:</p> <ul style="list-style-type: none"> • Amplitud: 0.8 para cada armónico. • Tiempo: equivalente a la señal original.
<p>Gráfico de Dispersión:</p>  <p>Ejes:</p> <ul style="list-style-type: none"> • Tipo → Bidimensional. • Eje X → Horizontal. • Eje Y → Vertical. 	<p>Ejes complementarios:</p> <ul style="list-style-type: none"> • Eje X → Espacial, canales de audio estéreo. • Eje Y → Frecuencia, valor máximo asociado a frecuencia más alta. <p>Características adicionales:</p> <ul style="list-style-type: none"> • Amplitud de la sinusoide: [0.2 – 1] para cada señal pura. • Tiempo: tiempo de sonificación escalado según distancia. <p>Características sinusoides armónicas:</p> <ul style="list-style-type: none"> • Amplitud: 0.8 para cada armónico. • Tiempo: equivalente a la señal original.

La Tabla 2.2, describe el cambio a realizarse de los ejes de coordenadas tradicionales hacía ejes auditivos, además de describir características adicionales como el tratamiento de los armónicos que se agregarán a la señal pura para modificar el timbre.

2.2.2.4 Transformar Datos de Origen a Frecuencia y Tiempo

Interpolarse datos desde frecuencia hacía amplitud es un proceso complejo y debe considerarse varios aspectos; inicialmente se debe manejar un rango de frecuencias correcto, de forma que se garantice el entendimiento de la interpretación como el bienestar en la salud de los usuarios. Los pasos posteriores dependerán del tipo de gráfico a ser implementado.

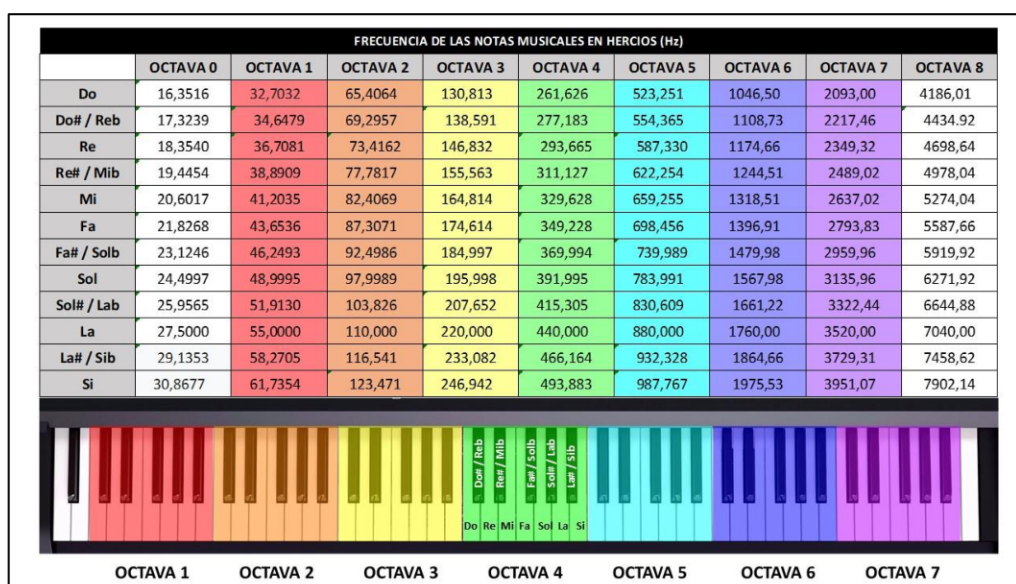


Figura 2.8: Frecuencias de las notas musicales [40].

Según se puede observar en la Figura 2.8, el rango de frecuencias de las notas musicales va desde los 16.35 [Hz] hasta los 7902.14 [Hz]. El rango a utilizar para todos los tipos de gráficos será el comprendido entre la nota musical RE hasta la nota musical LA (293.67 [Hz] – 1760 [Hz]). Los gráficos de distintos tipos requerirán tratamientos considerablemente distintos. Los pasos empleados se observan a continuación:

2.2.2.4.1 Gráfico de Línea

Para sonificar un gráfico se requiere interpolar valores desde datos originales que tienen una naturaleza particular hacía un rango de frecuencias que a su vez tienen una naturaleza distinta. El proceso realizado, mediante programación en *googlescript*, es el siguiente:

- a) Calcular el tiempo por segmento que debe asignarse al gráfico de línea en cuestión. Para calcular el tiempo por segmento se realiza el siguiente procedimiento:



Figura 2.9: Gráfico de línea referencial.

$$t_{segmento} = \left[\frac{d_1}{d_T} * t_{sonificación} ; \frac{d_2}{d_T} * t_{sonificación} \right] \quad (2.1)$$

Donde:

- $t_{segmento}$ → Corresponde a un vector que contiene los tiempos de todos los segmentos que conforman la línea.
- d_1 y d_2 → Corresponden a la magnitud del segmento.
- d_T → Corresponde a la magnitud total de la línea.
- $t_{sonificación}$ → Corresponde al tiempo ingresado por el usuario para sonificar.

Para calcular las magnitudes de cada segmento se utiliza la definición de distancia entre dos puntos y la magnitud total corresponde a la suma de todas las magnitudes de segmento. Por otra parte, el tiempo de sonificación es ingresado por el usuario.

- Escalar los datos originales de modo que no existan valores negativos. En caso de existir un valor negativo, se suma el menor de los valores detectados a todos los elementos de la matriz extraída en base al rango de celdas seleccionado.
- Normalizar los datos, de esta forma se obtiene un rango de valores de 0 a 1.
- Interpolar valores normalizados hacía valores de frecuencia, de modo que los valores máximos coincidan; es decir, la frecuencia máxima será asignada al dato que tenga el mayor valor de todo el conjunto de datos. El valor devuelto será un vector de frecuencias.

- e) Colocar las frecuencias calculadas en un vector y en otro vector independiente ubicar los tiempos de segmento. Posteriormente enviar los datos al cliente.

2.2.2.4.2 Gráfico de Barras

Los datos organizados mediante gráficos de barras tienen un comportamiento distinto al observado en gráficos de línea. En este caso particular, las barras expresan una magnitud, una dirección y una posición que permite diferenciar entre distintas barras. El procedimiento seguido, realizado mediante *googlescript*, es el siguiente:

- a) Extraer la barra cuyo valor absoluto sea mayor (barra con mayor magnitud).

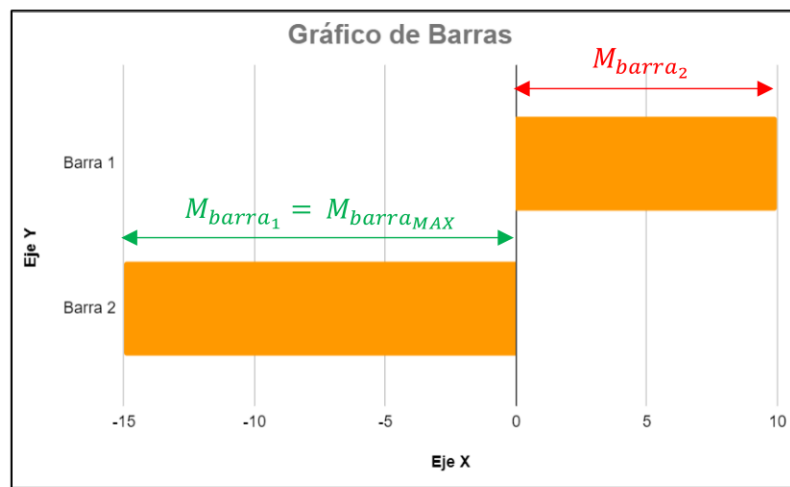


Figura 2.10: Gráfico de Barras Referencial.

- b) Extraer el tiempo de sonificación de barra, mismo que se calcula de la forma:

$$t_{barra} = \left[\frac{M_{barra_1}}{M_{barra_{MAX}}} * t_{sonificación} ; \frac{M_{barra_2}}{M_{barra_{MAX}}} * t_{sonificación} \right] \quad (2.2)$$

Donde:

- t_{barra} → Corresponde a un vector con los tiempos que sonará cada barra.
 - M_{barra_1} y M_{barra_2} → Corresponde a las magnitudes de las barras.
 - $M_{barra_{MAX}}$ → Corresponde a la barra cuya magnitud es mayor.
- c) Los espaciamientos entre barras en el Eje Y son los mismos, por tanto, se asignan valores de frecuencia correspondientes a una nota musical para cada barra. La barra con la ubicación más alta tendrá la nota con mayor frecuencia del rango

disponible, es decir nota musical LA. Las barras posteriores irán bajando su frecuencia según corresponda.

- d) Colocar las frecuencias y los tiempos de barra en dos vectores independientes. Posteriormente enviarlos al cliente.

2.2.2.4.3 Gráfico de Pastel

El gráfico de pastel es unidimensional únicamente expresando el porcentaje de la fila sobre el total de la matriz. Los pasos realizados en *googlescript*, se describen a continuación:

- a) Calcular el porcentaje asociado a cada rebanada, dividiendo el máximo para cada columna de dato.
- b) Calcular el tiempo de rebanada, mismo que se calcula de la forma.

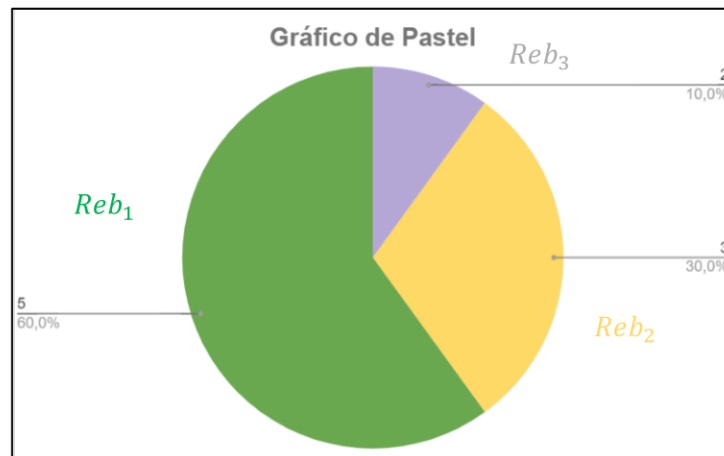


Figura 2.11: Gráfico de Pastel Referencial.

$$t_{rebanada} = [Reb_1 * t_{sonificación} ; Reb_2 * t_{sonificación}] \quad (2.3)$$

Donde:

- $t_{rebanada}$ → Corresponde a un vector con los tiempos que sonará cada rebanada.
 - Reb_1 y Reb_2 → Corresponde al valor de la rebanada, mismo que es un valor entre 0 y 1.
- c) Los valores de frecuencia son escalados empleando el mismo método del gráfico de barras. Se asigna la mayor frecuencia del rango (nota musical LA) a la rebanada con el mayor valor, por otra parte, las rebanadas con valores inferiores irán tomando el valor de una nota musical con menor frecuencia.

- d) Se coloca los valores de frecuencia y tiempos de rebanadas calculados en dos vectores independientes. Posteriormente, enviarlos al cliente.

2.2.2.4.4 Gráfico de Dispersión

El cálculo en cuanto a frecuencia es exactamente igual al observado en la descripción del gráfico de línea. Por otro lado, para conocer el tiempo por punto de la dispersión se utilizará la siguiente expresión:

$$t_{punto} = \frac{t_{sonificacion} \times (0.2)}{N_{puntos}} \quad (2.4)$$

Donde:

- N_{puntos} → Corresponde al número de puntos de la dispersión.

De esta forma, todos los puntos sonarán un mismo valor de tiempo. Además, se asegura que el tiempo total que suenen los puntos sea relevante, al corresponder al 20% del tiempo de sonificación total. Los tiempos por punto y frecuencia se colocarán en dos vectores independientes; posteriormente, estos serán enviados al cliente.

2.2.2.5 Espaciado de Audio (Panning)

El espaciado del audio o generación de audio espacial se aplicará en todos los gráficos bidimensionales, es decir el gráfico de pastel no tendrá un eje espacial, puesto que es un gráfico unidimensional. Debido a que la mayoría de los dispositivos actuales utilizan nada más 2 bocinas (sonido estéreo), se utilizarán 3 canales de audio derivados del mismo: izquierda, derecha y centro.

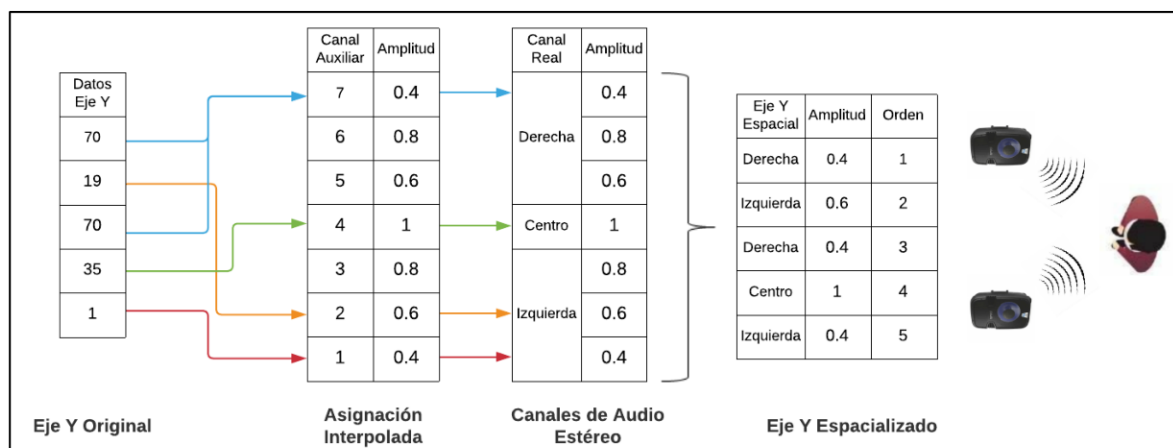


Figura 2.12: Funcionamiento proceso de espaciado de audio.

Según se puede observar en la Figura 2.12, el proceso de generación del eje de audio espacial funciona de la siguiente forma:

- Tomar datos del Eje Y de gráfico tradicional e interpolarlos en 7 rangos, que corresponden a canales de audio auxiliares, asignando una amplitud de senoide puntual. Se decidió utilizar 7 rangos tras realizar un estudio empírico de la posible cantidad de intervalos a ser utilizados.
- Transformar los canales de audio auxiliar hacia canales de audio utilizables por bocinas estéreo, manteniendo la amplitud.
- Devolver los datos transformados de modo que suenen en orden en los canales devueltos y con la amplitud determinada, solo puede sonar un canal de audio a la vez y el sonido se concatenará para evitar cortes en el audio.

La amplitud de la senoide se traduce en palabras comunes como el volumen del audio, según el proceso desarrollado. Si un dato se encuentra alejado de los demás hacia la izquierda o derecha su sonido será más suave, otorgando la idea de que la fuente de sonido se está alejando. Si bien el ejemplo de la Figura 2.12 se describe para datos sencillos del 0 al 70, el proceso desarrollado funciona para cualquier rango de datos.

El procesamiento final devolverá un vector con valores espaciales; es decir, con el canal correspondiente y la amplitud del canal. Finalmente, los valores de dicho vector deberán ser agrupados correctamente con su valor de frecuencia y tiempo correspondientes.

2.2.2.6 Montaje de audio sobre contexto de audio y osciladores

El procesamiento de los pasos anteriores fue desarrollado casi en su totalidad sobre *googlescript* en *Apps Script*, sin embargo, el sonido a realizarse debe ser montado sobre *javascript*, puesto que *googlescript* no puede manejar elementos propios del cliente, como audífonos o parlantes. De los procesamientos anteriores, el servidor devolverá tres vectores cuyo procesamiento fue descrito en los pasos anteriores. Los mencionados vectores corresponden a:

- Vector de frecuencias: se asigna una frecuencia por punto si se trata de un gráfico de línea o dispersión, una frecuencia por barra en gráficos de barras y una frecuencia por rebanada en el caso del gráfico de pastel.
- Vector de tiempos: se asigna un tiempo por segmento para gráficos de línea, un tiempo por barra para gráficos de barras, un tiempo por rebanada de gráfico de pastel o tiempo por punto de gráfico de dispersión.

- Vector espacial: se asigna un canal auxiliar y su respectiva ganancia para cada punto del gráfico de líneas y dispersión, y un vector de canales asociado a cada barra del gráfico de barras.

El contexto de audio es un objeto que se debe crear de manera global, de modo que cualquier función tenga acceso al mismo. Por otro lado, los osciladores deben ser creados de manera global y reiniciados cada que se realice una sonificación. En esencia el procesamiento a ser realizado es el siguiente por secciones, mismas que se describen a continuación:

```

1 // DEFINICIÓN BASE INICIAL: -----
2 //Creando contexto de audio
3 var audio_ctx = new (window.AudioContext || window.webkitAudioContext)();
4 //Creando oscilador global
5 var oscillator = audio_ctx.createOscillator();
6 oscillator.type = 'sine';
7 //Crear nodo de fusión de canales
8 let merger = audio_ctx.createChannelMerger(6);
9 merger.channelInterpretation = 'discrete';
10 //Creando ganancias locales
11 let ganancia_izq = audio_ctx.createGain();
12 let ganancia_der = audio_ctx.createGain();
13 let ganancia_cent = audio_ctx.createGain();
14 //Conectar generador de tonos con distintos canales de audio
15 oscillator.connect(ganancia_izq);
16 oscillator.connect(ganancia_der);
17 oscillator.connect(ganancia_cent);
18 //Conectar las amplitudes de canal a un único búfer de audio
19 ganancia_izq.connect(merger,0,0);
20 ganancia_der.connect(merger,0,1);
21 ganancia_cent.connect(merger,0,2);
22 }

```

Código 2.8: Codificación básica de una sonificación en frecuencia, tiempo, amplitud y espacio, definición de variables; sobre *javascript*

- a) Se comienza definiendo un contexto de audio y asociándole un oscilador. Estos elementos deben asociarse a variables globales⁵. Código 2.8, líneas 2 a 6.

⁵ Las variables globales son variables, que debido a su definición pueden ser utilizadas tanto dentro como fuera de funciones o bloques de programación aislados.

- b) Se crean distintas amplitudes por canal y estas se conectan al oscilador, de este modo se puede cambiar el sonido de canal fácilmente. Estas deberían ser variables locales⁶, para poder ser reiniciadas en cada ejecución. Código 2.8, líneas 11 a 17.
- c) Se crea un nodo de fusión (*merger*) y se asocia los sonidos de todos los canales en un único búfer de audio. Al igual que con las amplitudes por canal, este debe reiniciarse en cada ejecución por cuanto la variable debe ser local. Código 2.8, líneas 7 a 9 y 18 a 22.

Tras realizar la definición inicial de variables es necesario asignar los valores correctos al generador de tonos (oscilador), además de colocar dichos valores en el tiempo correcto. El proceso por realizarse es descrito a continuación:

```

1 // ASIGNACIÓN DE FRECUENCIAS: -----
2 oscillator.frequency.setValueAtTime(frecuencia, tiempo);

3 //Para gráficos de línea:
4 oscillator.frequency.linearRampToValueAtTime(frecuencia, tiempo);

5 //AMPLITUD DE SEÑAL SINUSOIDAL Y ESPACIAMIENTO DE AUDIO: -----
6 //Para gráficos de pastel y asignaciones de gráfico de línea
7 ganancia_izq.gain.setValueAtTime(ganancia, tiempo);

8 //Para gráficos de barras, líneas y dispersión:
9 ganancia_der.gain.linearRampToValueAtTime(ganancia, tiempo);

10 //CONECTAR CANALES DE AUDIO A LOS PARLANTES DEL ORDENADOR Y REPRODUCIR: --
11 merger.connect(audio_ctx.destination);
12 oscillator.start(0);
13 oscillator.stop(tiempo_final);

```

Código 2.9: Codificación básica de una sonificación en frecuencia, tiempo, amplitud y espacio, asignación de valores y reproducción.

- a) Para asignar una frecuencia es necesario realizar uno de los procesos siguientes, según amerite el caso:
- Asignar una frecuencia en un tiempo determinado. Código 2.9, líneas 1 y 2.
 - Provocar un cambio en la frecuencia que se encuentra asignado actualmente, hacia una frecuencia puntual en un tiempo determinado. Código 2.9, líneas 3 y 4.

⁶ Las variables locales describen variables, que debido a su definición pueden ser utilizadas exclusivamente dentro de funciones o bloques aislados de programación.

- b) Para asignar el canal y sus respectivos cambios, es necesario valerse de las amplitudes (ganancias) modificándolas según sea necesario.
- Asignar una ganancia en un tiempo determinado (Código 2.9, líneas 6 y 7)
Se destaca que mientras suene un canal, los demás deberían tener asignado como valor de ganancia 0, es decir no sonar.
 - Provocar un cambio en la ganancia que se encuentra asignada actualmente hacia un valor puntual en un tiempo determinado (Código 2.9, líneas 8 a 9).
Al igual que en proceso anterior, mientras un canal suene, los demás deben estar silenciados.
- c) El búfer de audio final, asociado al nodo de fusión, se asocia a los parlantes a los que tenga acceso el navegador. Código 2.9, línea 11.
- d) Finalmente, se reproduce el búfer de audio. Se detiene el sonido al final de la sonificación para evitar el uso innecesario de los parlantes del ordenador. Código 2.9, líneas 12 a 13.

El código completo de la implementación realizada a lo largo de las etapas anteriores es adjuntado en el Anexo IV.

2.2.2.7 Función de Compartición de Archivo

Si bien *Google Spreadsheets* cuenta con una funcionalidad de compartición de archivo, esta requiere de configurar permisos y establecer roles para los usuarios agregados. Se desarrolló una función para permitir compartir la hoja de cálculo únicamente introduciendo el correo de la persona con quién se desee compartir el archivo.

```
1 function compartir_archivo(correo){
2   let url = SpreadsheetApp.getActive().getUrl();
3   let archivo_actual = SpreadsheetApp.openByUrl(url);
4   let id = archivo_actual.getId();
5
6   try{
7     let mensaje = "Archivo de Spreadseet Compartido";
8     let usuario = {role: "writer", type: "user", value: correo};
9     Drive.Permissions.insert(usuario, id, {emailMessage: mensaje});
10    return JSON.stringify("CORRECTO");
11  }
12  catch (e) {
13    Logger.log(e);
14    return JSON.stringify("ERROR");
15  }
```

Código 2.10: Función para compartir documento, sobre *googlescript*.

En el Código 2.10, se utiliza la cuenta de *Google Drive* del usuario que tiene abierto el documento para enviar el mismo vía correo electrónico hacia un usuario en particular (líneas 6 a 9). El rol del usuario, con quién se comparte la hoja de cálculo, es configurado de modo que nuevos usuarios no requieren instalar el *add-on* y pueden utilizarlo directamente.

2.2.3 CODIFICACIÓN DE LA INTERFAZ

Según se definió en la Etapa de Diseño se ha decidido codificar dos interfaces distintas: una gráfica enfocada a ser utilizada por usuarios con discapacidad visual ayudándose de un lector de pantalla y una auditiva enfocada a ser usada por el mismo tipo de usuarios, de manera directa sin la intervención total de un lector de pantalla.

2.2.3.1 Interfaz Gráfica de Usuario

En primer lugar, se procedió a utilizar la hoja de estilos (*CSS*) proveniente de *Google* y enfocada en desarrolladores de *Apps Script* y similares; esto debido a que el espaciamiento y manejo de márgenes puede ser complejo de configurar y utilizando el mencionado elemento gráfico, se evitan ese tipo de problemas.

Se colocaron exclusivamente los recursos gráficos fundamentales para el procesamiento del *add-on* y se siguieron los parámetros definidos durante en los requerimientos no funcionales. La interfaz final realizada es la siguiente:



Figura 2.13: Secciones Implementadas durante el desarrollo de la interfaz gráfica.

La Figura 2.13 describe la interfaz gráfica implementada, en donde los elementos visuales siguieron los parámetros establecidos durante el planteamiento de requerimientos no funcionales. Se destaca las siguientes puntualizaciones:

- La sección de Barra de Nombres de *Spreadsheets* es obligatoria y no se puede suprimir o cambiar, salvo el texto.
- La sección de marca, también conocida como sección de *branding*, no cuenta con descripciones de texto alternativo, para evitar que su lectura confunda al usuario de las opciones reales. Esta zona se puede observar en la Figura 2.13.
- Las secciones de parámetros seleccionables y rellenables fueron separadas y etiquetadas, según se observa en la Figura 2.13.
- La única alternativa opcional es colocada al final de las opciones obligatorias.
- Los botones de acción se colocan tras presentarse todas las opciones del *add-on*, y aquel con funcionalidad opcional y menor jerarquía se coloca al final.
- Un elemento visual decorativo sin funcionalidad se coloca en la sección inferior, separada de todos los elementos funcionales para evitar confusiones.
- El tamaño de la vista HTML con respecto al de la ventana de *Google Spreadsheets* es de aproximadamente el 20% y no es posible aumentar el mismo, pues dicho parámetro está bloqueado para evitar afectar la funcionalidad de la hoja de cálculo.
- Se ocuparon elementos visuales básicos, como cuadros de texto, botones y menús de selección, elementos que son compatibles con lectores de pantalla.

2.2.3.2 Interfaz Auditiva de Usuario

Tal cual fue planteado a lo largo de la etapa de diseño, para asegurar que una aplicación web sea útil para un público no vidente, es necesario codificar una interfaz complementaria, que permita a los usuarios comunicarse con el *add-on* sin la necesidad de requerir la interfaz gráfica codificada.

Para poder definir un flujo complementario, que funcione de manera prácticamente exclusiva con el teclado, se requiere del uso del objeto "*window*" de *javascript*. Mediante su uso se puede referenciar a la pestaña actual, y programar eventos que se lleven a cabo sobre la misma, mediante ratón y teclado [30].

Existe un problema al momento de codificar atajos de teclado en *add-on* que trabajen con cualquier aplicación de *G-Suite*, como es el caso. Los comandos insertados solo funcionan si el cursor se encuentra sobre la vista *HTML* del *add-on*, según se observa en la Figura 2.14.

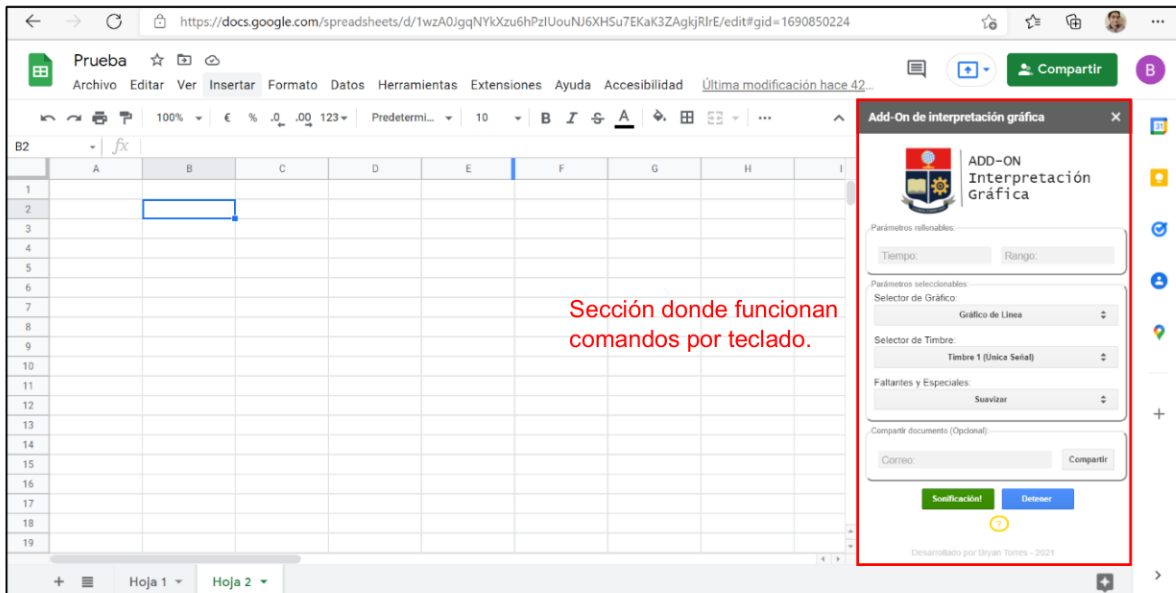


Figura 2.14: Sección donde se pueden manejar comandos por teclado y de manera general eventos sobre la pestaña.

Para solucionar el mencionado problema, se codifica una retroalimentación auditiva que funciona con el ratón. Cuando el usuario posee el ratón sobre el *add-on* se proveerá una retroalimentación positiva (sonido genérico de procesamiento correcto) y por otra parte si se sale del *add-on* se proveerá una retroalimentación auditiva negativa (sonido de error de *Windows XP*).

Por otro lado, se programaron eventos manejables por teclado, de modo que la forma de ejecución planteada en el Diagrama de Actividades, Apartado 2.1.5, Figura 2.4, se deba seguir de manera estricta y no sea posible desviarse. Se requiere confirmar todas las opciones y todos los pasos se pueden realizar exclusivamente empleando el teclado. Se ha requerido la programación de una funcionalidad de síntesis de voz (*text to speech*). La síntesis de voz hace posible realizar retroalimentaciones en todos los pasos, permitiendo el dictado de palabras variables tanto en semántica como longitud, además de una comunicación amigable con el usuario no vidente.

Los comandos por teclado realizados se describen a continuación:

- ENTER → Confirmar Selección.

- SHIFT + G → Comenzar sonificación, realizar una nueva sonificación.
- ALT + G → Detener evento auditivo. Detener sonificación completa o detener evento de síntesis de voz.
- ALT + P → Insertar parámetros rellenables. Como rango de celdas, tiempo o correo electrónico de compartición.
- SHIFT + F → Cambiar de opción en parámetros seleccionables. Como tipo de gráfico, tipo de timbre o manejo de valores faltantes.
- SHIFT + J → Comenzar sonificación una vez se rellenen todos los parámetros, volver a escuchar sonificación.
- SHIFT + H → Entrar en proceso de compartición de archivo mediante correo electrónico.

Se destaca que se verificó que los comandos no entren en conflicto con comandos del navegador o el sistema, además de que estos no generen edición en los datos. Adicionalmente, las letras que se suman a las teclas de comando se encuentran cercanas a aquellas teclas con identificadores táctiles, teclas F y J, para facilitar la inserción de los atajos por teclado

La funcionalidad de *text to speech* o síntesis de voz, se codificó empleando la Web Speech API, disponible en cualquier navegador actualizado. Sin embargo, la voz disponible variará de navegador a navegador. Se destacan las voces del navegador *EDGE* (voz por defecto de Bing y Azure) [31] y del navegador *Google Chrome* (una de las voces de *Google Translate*).

3 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES

Durante el presente Capítulo se mostrarán los resultados obtenidos tras finalizar la etapa de implementación del *add-on*, para lo cual se validan cada uno de los requerimientos funcionales y no funcionales, dentro de un entorno controlado.

3.1 RESULTADOS

3.1.1 ACTUALIZACIÓN DE TABLERO KANBAN

Se actualiza el tablero Kanban, mostrando la necesidad de completar pruebas y depuración del *add-on* desarrollado. Tras finalizar la tarea mencionada se procederá a dar por terminado el tablero.

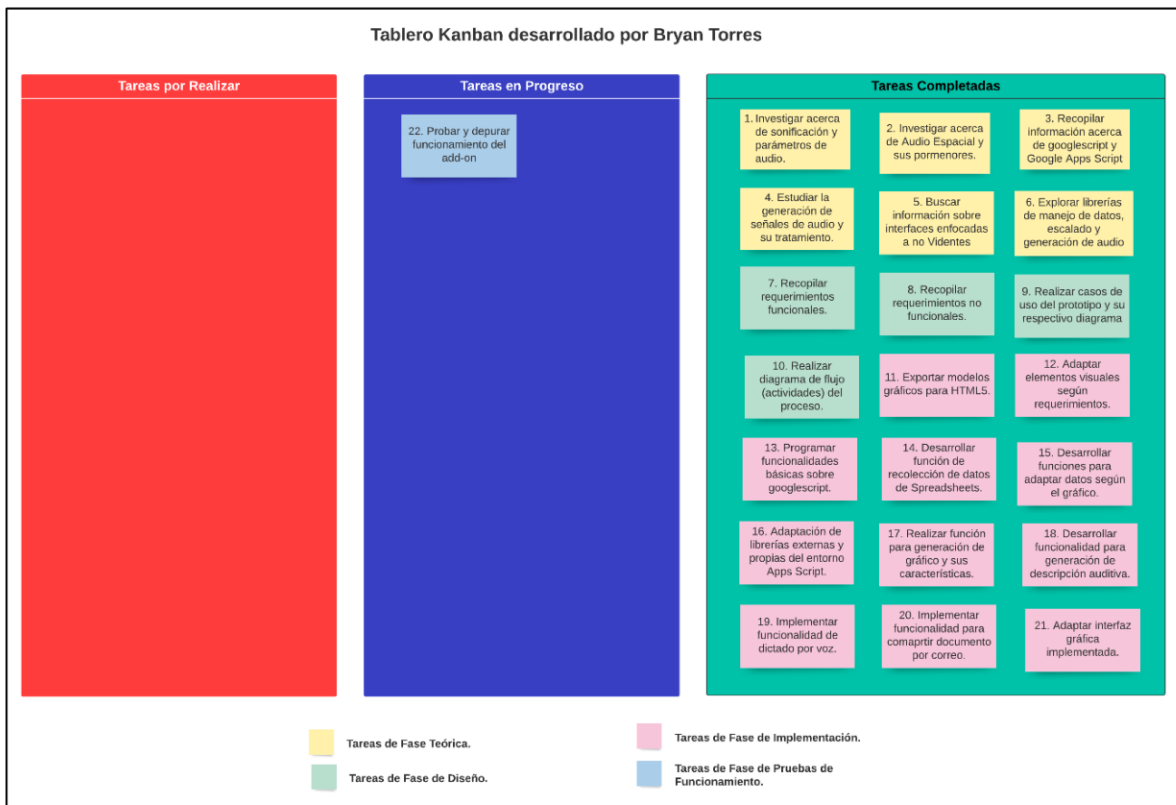


Figura 3.1: Actualización del tablero Kanban en Etapa de Pruebas y Depuración.

3.1.2 DEFINICIÓN DE ENTORNO DE PRUEBAS

Para realizar la comprobación del funcionamiento del prototipo, se procedió a crear un entorno de pruebas controlado, con 5 usuarios distintos. El entorno de pruebas comprendía las siguientes condiciones:

- Los usuarios corresponderán a usuarios sin discapacidad visual, sin embargo, utilizaron el *add-on* simulando ser individuos no videntes. Esta condición se consiguió mediante el uso de vendas de tela para evitar que los usuarios puedan mirar a la pantalla del ordenador.
- Previo a las pruebas los usuarios entraron en una etapa de inducción, donde se expusieron las distintas funcionalidades del *add-on*, los comandos necesarios para el uso del prototipo y pormenores de uso generales.
- Las pruebas se realizaron en modalidad telemática (virtual), mediante reuniones vía plataforma de *Zoom*; por otra parte, el *add-on* se compartió mediante la funcionalidad desarrollada en el mismo para compartir el documento, esto para evitar que los usuarios tengan que instalar el *add-on* mediante la copia de documentos en *Apps Script* y minimizar el tiempo necesario tanto para pruebas como para depuración.
- Los usuarios exploraron la interfaz tanto gráfica como auditiva, de modo que se pueda notar un contraste entra las posibles diferencias y las razones lógicas de diseño detrás de las mismas.
- Tras finalizar las pruebas del *add-on*, se realizaron preguntas de manera oral a los usuarios de modo que ofrezcan una retroalimentación con la experiencia aún en mente. Adicionalmente, se realizó un formulario de *Google Forms* donde se realizaron preguntas sencillas a los usuarios para verificar el cumplimiento de los requerimientos y contrastar con la retroalimentación oral ofrecida por los usuarios de prueba.
- Se llevaron a cabo dos reuniones, la primera el viernes 7 de enero del 2022, tras lo cual se tomó un tiempo de una semana para corrección de errores. De esta forma, la segunda sesión tuvo lugar el sábado 15 de enero del 2022. Durante la segunda reunión se verificó la corrección de los problemas encontrados.





Se usó la herramienta *Google Forms* debido a que permite generar estadísticas simples en tiempo real, según los resultados van siendo enviados por los usuarios. Los resultados de

la encuesta en *Google Forms*, serán adjuntados en el Anexo II, junto con el enlace de acceso al mismo.


3.1.3 VALIDACIÓN DE REQUERIMIENTOS FUNCIONALES

Las validaciones de los Requerimientos Funcionales se realizaron tomando como punto de referencia las respuestas de los usuarios en la encuesta realizada mediante *Google Forms*, así como la retroalimentación oral brindada por los usuarios al finalizar las sesiones de pruebas del prototipo:

Tabla 3.1: Verificación de requerimientos funcionales.

ID	Descripción	Estatus	Observación
RF1	Sobre flujo de ejecución y descripción hablada (<i>text to speech</i>).	Primera Reunión - Parcialmente cumplido. 	Problemática Encontrada: El 20% de los usuarios de prueba expresaron mediante <i>Google Forms</i> que la voz hablaba un poco lento, lo que podía entorpecer la experiencia. Eventualidad encontrada en <i>Google Chrome</i> .
		Segunda Reunión - Cumplido 	Medida Correctiva: se ajustó la tasa de palabras por minuto, aumentándola ligeramente.
RF2	Sobre sonidos de retroalimentación complementarios.	Primera y Segunda Reunión - Cumplido 	Observación: Los usuarios manifestaron claridad ante los sonidos de retroalimentación presentes al entrar o salir en el <i>add-on</i> y no presentaron recomendaciones adicionales.
RF3	Ingreso de datos compatible con <i>Google Spreadsheets</i> y sencillo de utilizar.	Primera Reunión - Parcialmente Cumplido. 	Observación: Se manejan tres métodos de selección de datos, uno de ellos emplea la notación <i>A1</i> y no todos los usuarios conocen la notación.



		Segunda Reunión - Cumplido. 	Medida Correctiva: La descripción auditiva de ingreso de datos explica de manera más puntual los 3 métodos. Adicionalmente, se incluyó una explicación de la notación A1 en la documentación web.
RF4	Diversidad de métodos de selección de datos a sonificar.	Primera y Segunda Reunión - Cumplido. 	Observación: Se poseen tres métodos de selección de datos, dos mediante la interfaz de <i>Google Spreadsheets</i> y uno con la interfaz de usuario propia del <i>add-on</i> .
RF5	Elección de parámetros seleccionables sencilla. Correspondientes a tipo de gráfico, timbre.	Primera y Segunda Reunión - Cumplido. 	Observación: La elección de parámetros seleccionables está claramente diferenciada por interfaces, se tiene un método de selección mediante interfaz gráfica y otro mediante interfaz auditiva. Los usuarios demostraron conformidad con los métodos empleados.
RF6	Presencia de un método para sortear los valores faltantes O Inconsistentes.	Primera y Segunda Reunión - Cumplido. 	Observación: Se proveyeron de dos métodos de corrección de errores faltantes, ignorar faltantes o rellenar con el promedio de la agrupación. Los usuarios expresaron preferir el método de ignorar los valores faltantes, pues no se afectan los datos. Sin embargo, se decidió colocar un método alternativo para proveer más opciones al usuario.
RF7	Presencia de un método sencillo para compartir el documento con colaboradores.	Primera Reunión - Parcialmente cumplido. 	Observación: Se proveyó al usuario la opción de compartir el documento mediante correo electrónico y embeber el proceso dentro de la interfaz del <i>add-on</i> .
		Segunda Reunión -	Medida Correctiva: Se observó la necesidad de un método de comprobación de envío correcto del correo.





		Cumplido. 	Se proveerá una retroalimentación auditiva en caso de ser un envío correcto. Inicialmente se decidió otorgar permisos de revisión a los usuarios con quienes se compartía. Los usuarios de prueba demostraron la necesidad de otorgar permisos de edición a las personas con quienes se comparte, por tanto, se modificó el permiso hacía edición.
--	--	--	---




3.1.4 VALIDACIÓN DE REQUERIMIENTOS NO FUNCIONALES



Los Requerimientos no Funcionales corresponden a aquellos complementarios, que favorecen la interacción con el usuario y brindan una experiencia correcta. Al igual que con los Requerimientos Funcionales, se pidió retroalimentación de los usuarios de prueba, de modo que las comprobaciones se realizaron basándose en las respuestas ofrecidas de manera oral tras terminar las pruebas y se complementan dichas respuestas con las recibidas por parte de los usuarios en el formulario generado mediante *Google Forms*.

Tabla 3.2: Verificación de requerimientos no funcionales.

ID	Descripción	Estatus	Observación
RNF1	Orden de la interfaz gráfica y uso de efectos visuales básicos.	Primera y Segunda Reunión - Cumplido. 	Observación: Se procuró colocar exclusivamente los elementos necesarios en la interfaz, adicionalmente se siguieron los estamentos planteados para diseño de interfaces basadas en <i>ETI</i> , Tabla 2.1, mismos que se basan en las recomendaciones proporcionadas por la <i>WCAG</i> . Los usuarios no manifestaron problemas con la estructura del <i>add-on</i> , su disposición, elementos o efectos visuales.
RNF2	Eficiencia de procesamiento, tiempos de ejecución.	Primera y Segunda Reunión - Cumplido 	Observación: Se limitaron los tiempos de procesamiento al máximo. Sin embargo, la instancia de <i>Google Cloud</i> es aleatoria por cuanto es posible que exista una diferencia entre una ejecución y otra. Si bien los tiempos de ejecución no son exagerados, se debe considerar una

			pequeña variación, causada por la comunicación con <i>Google Cloud</i> .
RNF3	El flujo de ejecución debe ser lógico y ordenado	Primera y Segunda Reunión - Cumplido. 	Observación: Los usuarios manifestaron conformidad con el flujo de ejecución.
RNF4	Capacidad máxima de usuarios 5 individuos simultáneamente.	Primera y Segunda Reunión - Cumplido. 	Observación: La cantidad de usuarios disponibles por documento es realmente variable y podría llegar a más de 5 individuos. Aún con lo planteado, se recomienda ser usado por una cantidad inferior a 5 individuos a la vez. Se detalla que la ejecución del <i>add-on</i> será individual de cada individuo, de modo que los demás usuarios no vean interrumpido su trabajo.
RNF5	Compatibilidad con al menos dos navegadores.	Primera y Segunda Reunión - Cumplido. 	Observación: Las <i>APIs</i> de <i>Web Audio</i> y <i>Web Speech</i> son propias de <i>javascript</i> y se asocian al navegador. Fue necesario verificar que estas dos <i>APIs</i> funcionen correctamente. Los usuarios de prueba utilizaron en su mayoría <i>Google Chrome</i> , sin embargo, el 20% de los usuarios utilizó <i>Edge</i> . Adicionalmente, las pruebas de desarrollo se realizaron sobre <i>Edge</i> y <i>Opera</i> , de modo que se garantiza la compatibilidad en más de 2 navegadores.
RNF6	Evitar saturar al usuario no vidente con demasiados eventos de audio a la vez.	Primera Reunión - No cumplido 	Problemática Encontrada: El 40% de los usuarios de prueba denotaron requerir de un mayor espaciamiento entre las explicaciones habladas mediante el sintetizador de voz. Adicionalmente, los usuarios de prueba recomendaron explicar el evento de reconocimiento de presencia del usuario sobre el <i>add-on</i> durante la presentación. Adicionalmente, el evento de bienvenida podía chocar con el evento realizado para

			comprobar si el usuario se encuentra sobre el <i>add-on</i> .
		Segunda Reunión - Cumplido. 	Medidas Correctivas: Se revisaron los eventos de audio que verifican si el usuario se encuentra o no sobre el <i>add-on</i> , se crearon contenciones evitando que este evento se ejecute durante la presentación del <i>add-on</i> o durante las sonificaciones, además se añadió una pequeña explicación durante la presentación para que los usuarios tengan claro el evento de presencia sobre el <i>add-on</i> . Se añadió silencios entre varios eventos de audio, de modo que las explicaciones habladas mediante síntesis de voz no choquen con otros eventos auditivos, interrumpan la sonificación o causen conflictos con la presentación.
RNF7	Simplicidad de opciones y atajos de teclado limitados a la cantidad necesaria.	Primera Reunión - Parcialmente Cumplido. 	Observación: Los eventos de teclado fueron limitados al mínimo posible y se buscó que eventos similares utilicen un mismo comando. El 20% de los usuarios expresaron que puede ser algo complejo encontrar las teclas con los ojos vendados. Sin embargo, confirman que es correcto su ubicación cercana a las teclas identificadas por mecanografía, teclas F y J. Adicionalmente, se destaca que los usuarios no tenían experiencia previa simulando situaciones por el estilo.
		Segunda Reunión - Cumplido 	Medidas correctivas de soporte: Se verificó que previo a un evento que requiera la inserción de un comando por teclado, se tenga una retroalimentación hablada mediante síntesis por voz, de manera que recuerde a los usuarios que teclas presionar en cada sección.
RNF8	Manejo de permisos de la hoja de cálculo	Primera y Segunda Reunión	Observaciones: Se verificó los permisos que requirieron los usuarios para la ejecución del <i>add-on</i> . Es

	Google Spreadsheets.	- Cumplido 	así como los permisos se limitaron a los mínimos necesarios; acceso a modificación del archivo, acceso a <i>Google Drive</i> , acceso al correo y nombre del usuario. Previo a la realización de prueba se eliminaron servicios de <i>Google</i> que consumían permisos adicionales y que fueron usados durante la etapa de implementación, pero se descartaron durante la misma. Se verificó que no se almacene ningún tipo de información personal del usuario. Tras cerrar la hoja de cálculo, todas las variables o datos del <i>add-on</i> son eliminados y estos solo están disponibles mientras el usuario use una funcionalidad que lo requiera. Por ejemplo, acceso <i>Google Drive</i> e información de correo si se requiere compartir el documento.
RNF9	Tamaño del <i>add-on</i> en un rango no mayor al 30% del tamaño de la ventana o menor al 5% del mencionado tamaño.	Primera y Segunda Reunión - Cumplido 	Observación: En los <i>add-on</i> sobre <i>Google Spreadsheets</i> no es posible modificar el tamaño, el tamaño aproximado con respecto a la ventana del navegador es de un 20%, con respecto al total. Apps Script permite modificar el tamaño de las vistas <i>HTML</i> en otro tipo de implementaciones como complementos para <i>Gmail</i> .
RNF10	Ejecutar el <i>add-on</i> no requiere de salir de la pestaña de <i>Google Spreadsheets</i> .	Primera y Segunda Reunión - Cumplido 	Observación: La llamada del <i>add-on</i> se realiza mediante una de las pestañas propias de <i>Google Spreadsheets</i> llamada "Extensiones" y se colocó una opción llamada "Add-On SPSH", de esta forma, se verifica que el usuario no tenga que realizar procedimientos complejos para acceder a las funciones del prototipo. Los usuarios de prueba no presentaron inconvenientes o recomendaciones con respecto a este requerimiento funcional.
RNF11	Rango audible de sonificación correcto y que no	Primera y Segunda Reunión	Observación: Para verificar el rango audible, se tomaron frecuencias en el mismo rango de las notas

	<p>genere perjuicios en el usuario no vidente.</p>	<p>- Cumplido.</p> <p style="text-align: center;">✔</p>	<p>musicales entre la cuarta y sexta escala. Adicionalmente, se proveyó la opción de cancelar la sonificación en caso de que el usuario tenga un conflicto con el audio generado.</p> <p>Los usuarios no presentaron conflictos con las sonificaciones realizadas.</p>
--	--	---	--

La Tablas 3.1 y 3.2 describen los procedimientos realizados para verificar el cumplimiento de los requerimientos planteados durante la Etapa de Diseño. Fue necesario la corrección de varios módulos para el cumplimiento total de los mismos, así mismo, se incluyeron correcciones complementarias surgidas de las experiencias obtenidas con los usuarios de prueba o de recomendaciones planteadas por ellos.

3.1.5 CIERRE DEL TABLERO KANBAN

Una vez realizadas las pruebas sobre el prototipo y depurar los problemas encontradas durante su ejecución, se procede a cerrar el tablero Kanban, por cuanto se finalizaron todas las tareas planteadas, según se observa en la Figura 3.2.

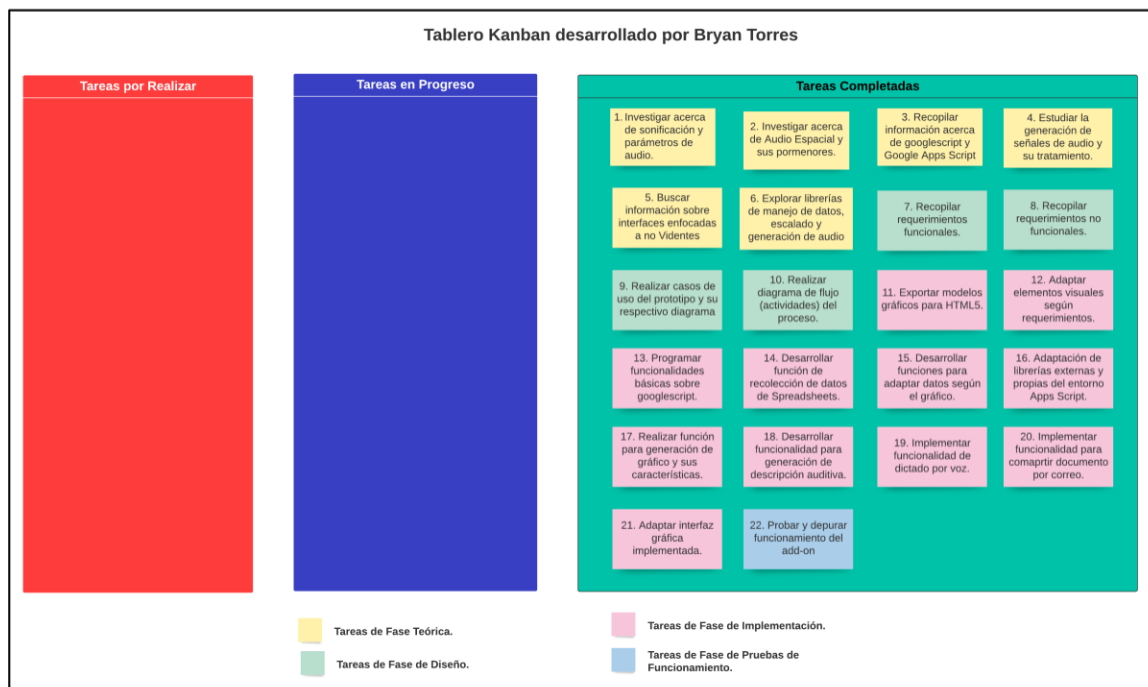


Figura 3.2: Finalización del Tablero Kanban.

3.2 CONCLUSIONES Y RECOMENDACIONES

3.2.1 CONCLUSIONES

El presente Trabajo de Integración Curricular fue desarrollado basándose en cuatro fases fundamentales: la Fase Teórica, de Diseño, de Implementación y de Pruebas de Funcionamiento. Derivado del proceso realizado durante los Apartados anteriores, se desprenden las siguientes conclusiones:

- Mediante el desarrollo de un *add-on* sobre la plataforma *Apps Script* se ha logrado generar una herramienta capaz de proveer una interpretación auditiva correspondiente a los gráficos de línea, barras, pastel y dispersión generados por la herramienta de gestión de hojas de cálculo *Google Spreadsheets*. Se utilizaron técnicas de sonificación y audio espacial enfocadas en la transformación de coordenadas tradicionales (x, y, z) hacia coordenadas complementarias expresables mediante descripciones auditivas. La correcta interpretación de la información obtenida en la fase teórica permitió un desarrollo equilibrado que mediante al uso de servicios y *APIs* propios de *Google*. Conjuntamente con las *APIs* de *Web Audio* y *Web Speech*, hicieron posible la realización de un prototipo de sonificación de gráficos eficiente y apto para el uso de usuarios no videntes.
- El estudio del entorno *Apps Script* y su integración con *Google Spreadsheets* fue complejo y desembocó en el análisis y entendimiento en primer lugar de su arquitectura y el manejo de peticiones con *Google Cloud Platform* y en segunda instancia de la integración de este ecosistema en con el mundo de desarrollo web tradicional, manejado mediante lenguaje javascript. Sortear los problemas propios de esta integración requirió de la búsqueda de opciones particulares, debido principalmente a problemas de compatibilidad, sin embargo, se logró crear un equilibrio y desarrollar un *add-on* que cumple con los objetivos planteados y sigue altos estándares de calidad.
- El cumplimiento cabal de las actividades planteadas permitió el desarrollo de un prototipo eficiente, además de ayudar a identificar problemas y buscar mejores alternativas a procedimientos planteados inicialmente. Mediante el uso de la metodología Kanban, se verificó la finalización de las diversas etapas de manera ordenada y se culminó el desarrollo de manera satisfactoria.
- Las distintas funciones del *add-on* fueron realizadas a través del uso correcto de las diversas tecnologías disponibles y de la base teórica, tras su uso. Se consiguió

desarrollar todas las funcionalidades planteadas y hacer que las mismas sean consistentes e incluso mejores con respecto a las planteadas idealmente en un inicio.

- La realización de pruebas en un entorno controlado fue fundamental para la culminación del desarrollo del prototipo. El análisis de la retroalimentación oral brindada por los usuarios de prueba permitió corregir problemáticas y principalmente mejorar funcionalidades implementadas, que, si bien cumplían sus objetivos, pudieron adaptarse de mejor manera para ser más amigables y eficientes con el usuario. Adicionalmente, los resultados obtenidos por la encuesta realizada sobre *Google Forms*, devolvió resultados estadísticos que permitieron encontrar eventualidades adicionales corrigiendo problemas encontrados y mejorando procesos que lo necesitarán.
- El lenguaje *googlescript* sobre el cual se programa toda la lógica de gestión de la aplicación *Google Spreadsheets*, puede ser limitada y a pesar de estar basado en *javascript*, presenta incompatibilidad y funcionamientos inadecuados con funciones de su lenguaje padre. El desarrollo de funciones y procesos sobre el mencionado lenguaje debe realizarse de manera meticulosa y sus resultados deben ser depurados continuamente.
- Generar sonificaciones correctas, requiere de un procesamiento ordenado de los datos de origen. Es importante comprender, que la información recolectada de los usuarios tendrá una naturaleza puntual, y, por tanto, la adaptación de estos datos a parámetros de audio requerirá de procesos tales como reescalamiento, normalización, interpolación, entre otros. La realización de este proceso es fundamental para la generación de interpretaciones apegadas a la realidad y por tanto útiles para los usuarios.
- El segmento minoritario de personas con discapacidad visual ha sido relegado durante décadas del uso de herramientas tecnológicas y de ofimática; la presencia de herramientas del tipo de lectores de pantalla permite la integración de este segmento de la población con el uso de ordenadores y demás. Es fundamental que el desarrollo web tenga un enfoque de accesibilidad y que los nuevos sitios web tengan herramientas, tales como el uso de "*alternative text*" y la aplicación de recomendaciones tales como *WCAG* o *ETI*.

- Es recomendable en lo posible evitar el uso de librerías complejas basadas en *javascript*. Si bien es posible configurar funciones en *javascript*, todos los procesos están hospedados en los servidores de *Google* y esto causa que muchas librerías no tengan un correcto funcionamiento o directamente sean inservibles.
- La lógica programada sobre *googlescript*, de la presentación programada sobre *HTML* están lógicamente separadas, es por ello que al intentar pasar variables de un lado hacia el otro será necesario el paso de las mismas encapsuladas como objetos *JSON*.

3.2.2 RECOMENDACIONES

Durante las distintas fases, se presentaron varios problemas que necesitaron de soluciones de diversa índole, en base a las eventualidades encontradas se plantean las siguientes recomendaciones:

- Durante el desarrollo de *add-ons* sobre *Apps Script*, se recomienda separar la parte lógica en *googlescript* de la parte lógica en *javascript* y esto a su vez de las partes visuales, más bien enfocadas al cliente, del tipo de hojas de estilos (*CSS*) y la vista *HTML* como tal. Para realizar esta tarea es necesario crear una función que permita conectar la interfaz de usuario con las funciones externas; si bien esto puede ser algo complejo, su desarrollo permite separar correctamente las secciones lógicas de las gráficas.
- Aquellos eventos que requieran asociarse a la vista *HTML* deberán ser programados en *javascript*, puesto que la lógica manejada por *googlescript* no contempla el uso de elementos del tipo (*window*) o similares, por tanto, no es posible hacer una referencia a la vista *HTML* como tal.
- Dentro de la documentación oficial de *Google* se provee una hoja de estilos (*CSS*) que puede ser referenciada desde la vista *HTML*, se recomienda utilizar dicho elemento. Es relativamente complicado manejar las posiciones de los objetos gráficos sobre la mencionada vista, pero el uso de esta hoja de estilos permite configurar rápidamente los márgenes, espaciamientos, posiciones e incluso animaciones.
- Es una buena práctica de desarrollo web siempre utilizar un seccionamiento estándar, pues esto permite que usuarios con discapacidad visual puedan adaptarse rápidamente a los sitios web. Esto a su vez aumenta nuestra posible

cuota de mercado, además de permitir que usuarios videntes tengan una idea clara de la ubicación de los objetos y favorecer su navegación.

- En un inicio se planteó la generación de contextos de audio locales, de modo que estos sean creados cada vez que sea necesario y sea imposible el conflicto entre funciones debido al uso de variables globales. Sin embargo, debido a la naturaleza desechable de los osciladores (generadores de tono) se recomienda utilizar un único contexto de audio, asociado a la pestaña general y cada vez que sea generada una sonificación reasignarlo, es decir reiniciarlo. Esto permite eliminar eficientemente las características definidas en los osciladores y facilita el desarrollo de funciones que permitan detener la sonificación en curso.

Adicionalmente, en base a la experiencia adquirida durante el desarrollo del presente prototipo se presentan posibles recomendaciones enfocadas a desarrollos futuros sobre esta plataforma:

- Se podría definir un método más eficiente para el uso de llamadas asíncronas, es decir funciones que se llevan a cabo cuando el usuario efectúa algún evento. En futuros desarrollos es posible explorar el uso de concatenaciones de eventos de tipo “*SetTimeout*”, opción común en desarrollo web tradicional. Actualmente los eventos “*SetTimeout*” no funcionan correctamente en *Apps Script*, por cuánto se puede buscar una solución desarrollada en dos fases, de manera parcial en *googlescript* y agregar una sección complementaria en *javascript*.
- Se podría asignar de la amplitud de la senoide a un elemento dinámico, como un *slider*, de modo que se pueda modificar en cada sonificación el timbre de sonificación. Se plantea como una alternativa experimental, sin embargo, se recomienda alejarla del manejo del usuario o colocar un rango de amplitud limitado, puesto que se puede generar un sonido poco agradable. Adicionalmente, en futuros proyectos se podrían agregar opciones para modificar más específicamente campos de la sonificación como: rangos de frecuencias, manejo de tiempos, etc.
- Se desarrolló la interpretación de gráficos bien diferenciados. En un futuro, se puede llegar a realizar la sonificación de gráficos mixtos, por ejemplo, un gráfico que contiene una barra, una línea y un pastel.

4 REFERENCIAS BIBLIOGRÁFICAS

- [1] WAI (Web Accessibility Initiative), «Web Content Accessibility Guidelines (WCAG) Overview,» WAI, 6 Diciembre 2021. [En línea]. Available: <https://www.w3.org/WAI/standards-guidelines/wcag/#versions>. [Último acceso: 16 Enero 2022].
- [2] Mozilla Developer Network Web Docs, «Web Audio API,» Mozilla, 4 Enero 2022. [En línea]. Available: https://developer.mozilla.org/es/docs/Web/API/Web_Audio_API. [Último acceso: 6 Enero 2022].
- [3] Mozilla Developer Network Web Docs, «Using the Web Speech API,» Mozilla, 5 Enero 2022. [En línea]. Available: https://developer.mozilla.org/es/docs/Web/API/Web_Speech_API/Using_the_Web_Speech_API. [Último acceso: 6 Enero 2022].
- [4] B. Walker y M. Nees, Principles of Sonification: An Introduction to Auditory Display and Sonification, Atlanta, Georgia, USA: Georgia Institute of Technology, 2006.
- [5] H. Kaper, S. Tipei y W. Elizabeth, «Data sonification and sound visualization, in Computing in Science & Engineering,» *IEEE Explore*, vol. 1, n° 4, p. 50, 30 Agosto 1999.
- [6] S. Lodha, J. Beahan, T. Heppe, A. Joseph y Z.-U. Brett, «MUSE: A Musical Data Sonification Toolkit,» *4th International Conference on Auditory Display (ICAD)*, vol. 1, n° 1, p. 2, 1997.
- [7] G. Gibson, «Overtone Series, Addition of Waves and Tone Quality,» University of Connecticut, The Physics of Music Subject Content, 10 Octubre 2010. [En línea]. Available: https://www.phys.uconn.edu/~gibson/Notes/Section4_2/Sec4_2.htm. [Último acceso: 5 Enero 2022].
- [8] P. Morgado, V. Nuno, T. Langlois y O. Wang, «Self-Supervised Generation of Spatial Audio for 360° Video,» *The Computer Research Repository*, vol. abs/1809.02587, n° 10, pp. 1-2, 2018.
- [9] The International Telecommunication Union (ITU) Assembly, «RECOMMENDATION ITU-R BS.775-3 - Multichannel stereophonic sound system with and without accompanying picture,» ITU Radiocommunication, Ginebra, Suiza, 2012.
- [10] Google , «About Sheets,» Google, 6 Enero 2022. [En línea]. Available: <https://www.google.com/intl/es/sheets/about/>. [Último acceso: 6 Enero 2022].
- [11] Google for Developers, «Google Apps Script,» Google, 1 Enero 2022. [En línea]. Available: <https://developers.google.com/apps-script>. [Último acceso: 5 Enero 2022].
- [12] Google for Developers, «Google Cloud Platform Projects,» Google, 4 Septiembre 2021. [En línea]. Available: <https://developers.google.com/apps-script/guides/cloud-platform-projects>. [Último acceso: 6 Enero 2022].

- [13] Google for Developers, «Script Projects,» Google, 11 Agosto 2021. [En línea]. Available: <https://developers.google.com/apps-script/guides/projects>. [Último acceso: 6 Enero 2022].
- [14] Soporte de Google, «Use App Scripts complements,» google, 10 Diciembre 2021. [En línea]. Available: <https://support.google.com/docs/answer/2942256?hl=es-419&co=GENIE.Platform%3DAndroid>. [Último acceso: 6 Enero 2022].
- [15] Mozilla Developer Network Web Docs, «AudioContext,» Mozilla, 14 Septiembre 2021. [En línea]. Available: <https://developer.mozilla.org/en-US/docs/Web/API/AudioContext>. [Último acceso: 6 Septiembre 2022].
- [16] Mozilla Developer Network Web Docs, «OscillatorNode,» Mozilla, 14 Septiembre 2021. [En línea]. Available: <https://developer.mozilla.org/en-US/docs/Web/API/OscillatorNode>. [Último acceso: 6 Enero 2022].
- [17] Mozilla Developer Network Web Docs, «ChannelMergerNode,» Mozilla, 14 Septiembre 2021. [En línea]. Available: <https://developer.mozilla.org/en-US/docs/Web/API/ChannelMergerNode>. [Último acceso: 6 Enero 2022].
- [18] Mozilla Developer Network Web Docs, «SpeechSynthesis,» Mozilla, 16 Diciembre 2021. [En línea]. Available: <https://developer.mozilla.org/en-US/docs/Web/API/SpeechSynthesis>. [Último acceso: 6 Enero 2022].
- [19] M. D. Arango Serna, L. F. Campuzano Zapata y J. A. Zapata Cortes, «Mejoramiento de procesos de manufactura utilizando Kanban,» *Ingenierías Universidad de Medellín*, vol. 14, n° 27, pp. 223-225, 2015.
- [20] Lucid Software Inc., «Sobre LucidSpark,» Lucid Software, 12 Enero 2022. [En línea]. Available: https://lucidspark.com/es/landing?utm_source=google&utm_medium=cpc&utm_campaign=_spark_es_tier2_mixed_search_brand_exact_&km_CPC_CampaignId=11288850210&km_CPC_AdGroupId=111099351095&km_CPC_Keyword=lucidspark&km_CPC_MatchType=e&km_CPC_ExtensionID=&km_CPC_N. [Último acceso: 15 Enero 2022].
- [21] E. Pontes, A. Guelfi y S. Kofuji, *E-Learning: Long-Distance and Lifelong Perspectives*, São Paulo, Brasil: IntechOpen, 2012.
- [22] Soporte de Google, «Edit spreadsheets with a screen reader,» Google, 1 Enero 2022. [En línea]. Available: <https://support.google.com/docs/answer/1632199?hl=en>. [Último acceso: 14 Enero 2022].
- [23] C. Enders, *Applied Missing Data Analysis*, New York, USA: Guilford Press, 2010.
- [24] Lucid Software Inc, «Sobre LucidChart,» Lucid Software, 1 Enero 2022. [En línea]. Available: https://www.lucidchart.com/pages/es/landing?utm_source=google&utm_medium=cpc&utm_campaign=_chart_es_tier2_mixed_search_brand_exact_&km_CPC_CampaignId=1501207859&km_CPC_AdGroupId=63362176052&km_CPC_Keyword=lu

cidchart&km_CPC_MatchType=e&km_CPC_ExtensionID=&. [Último acceso: 15 Enero 2022].

- [25] Cátedra de Diseño de Interfaces - Máster en Aplicaciones Multimedia, «¿Qué es una interfaz?,» Universitat Oberta de Catalunya, 10 Septiembre 2017. [En línea]. Available: <https://multimedia.uoc.edu/blogs/dii/es/que-es-una-interficie/>. [Último acceso: 9 Enero 2022].
- [26] S. Leuthold, J. Bargas-Avila y K. Opwis, «Beyond web content accessibility guidelines: Design of enhanced text user interfaces for blind internet users,» *International Journal of Human-Computer Studies*, vol. 4, nº 66, pp. 257-259, 2008.
- [27] Mozilla Developer Network Web Docs, «HTML: A good basis for accessibility,» Mozilla, 10 Octubre 2021. [En línea]. Available: <https://developer.mozilla.org/en-US/docs/Learn/Accessibility/HTML>. [Último acceso: 9 Enero 2022].
- [28] E. Rufino de Souzaa y S. Fernandes de Freitas, «Recommendations for improving interaction of blind people with the web,» *IOS Press*, vol. 41, nº 1, pp. 3443-3445, 2012.
- [29] J. Lindström, «Technological Solutions for Visually Impaired People in Sweden,» *Journal of Visual Impairment & Blindness*, vol. 10, nº 84, pp. 513-514, 1990.
- [30] Mozilla Developer Network Web Docs, «Window,» Mozilla, 7 Enero 2022. [En línea]. Available: <https://developer.mozilla.org/es/docs/Web/API/Window>. [Último acceso: 10 Enero 2022].
- [31] Microsoft Azure, «Features of Text to Speech,» Microsoft, 10 Enero 2022. [En línea]. Available: <https://azure.microsoft.com/es-es/services/cognitive-services/text-to-speech/#features>. [Último acceso: 22 Enero 2022].
- [32] R. Panda, R. Malheiro y R. Paiva, «Novel audio features for music emotion recognition,» *IEEE Transactions on Affective Computing*, vol. 10, nº 1109, pp. 1-6, 2018.
- [33] Google, «G-Suite - Google Workspace,» Google, 20 Enero 2022. [En línea]. Available: https://workspace.google.com/intl/es-419/?utm_source=google&utm_medium=cpc&utm_campaign=latam-T1-all-es-dr-bkws-all-all-trial-e-dr-1011272-LUAC0012555&utm_content=text-ad-none-any-DEV_c-CRE_479487543575-ADGP_Hybrid+%7C+BKWS+-+EXA+%7C+Ttxt+~+G+Suite-KWID_43. [Último acceso: 23 Enero 2022].
- [34] Dolby Laboratories Inc, «What is the LFE channel?,» Dolby Digital, 10 Octubre 2015. [En línea]. Available: https://www.dolby.com/uploadedFiles/Assets/US/Doc/Professional/38_LFE.pdf. [Último acceso: 23 Enero 2022].
- [35] Future Music Corporation, «Transitorios, modelado para diseño sonoro,» Future Music Corporation, 21 Enero 2015. [En línea]. Available: <http://www.futuremusic-es.com/transitorios-modelado-para-diseno-sonoro/>. [Último acceso: 22 Enero 2022].

- [36] C. Cullen, *The Sonic Representation of Mathematical Data*, Doctoral Theses, Dublin, Irlanda: Technological University Dublin, 2006.
- [37] A. Gajardo, «Consecuencias del Contenido Armónico en Motores Eléctricos,» *Motores y Generadores*, 22 Septiembre 2015. [En línea]. Available: <https://motoresygeneradores.com/consecuencias-del-contenido-armonico-en-motores-electricos/>. [Último acceso: 5 Enero 2022].
- [38] M. E. Árevalo Lizardo, «Introducción al Patrón de Arquitectura por Capas,» *Blog Personal*, 2 Diciembre 2010. [En línea]. Available: <https://arevalomaria.wordpress.com/2010/12/02/introduccion-al-patron-de-arquitectura-por-capas/>. [Último acceso: 9 Enero 2022].
- [39] Google, «About Drive,» *Google*, 9 Enero 2022. [En línea]. Available: <https://www.google.com/intl/es/drive/>. [Último acceso: 9 Enero 2022].
- [40] Equipo de Ciudad Pentagrama, «Clase 23: Tabla de Frecuencias Notas Musicales,» *Ciudad Pentagrama*, 28 Enero 2020. [En línea]. Available: <https://www.ciudadpentagrama.com/2020/01/tabla-frecuencias-notas-musicales.html>. [Último acceso: 9 Enero 2022].
- [41] Google Chrome Web Store, «Screen Reader,» *Google*, 27 Julio 2021. [En línea]. Available: <https://chrome.google.com/webstore/detail/screen-reader/kgejglhpjiefppelpmljglcjbhoiplfn>. [Último acceso: 15 Enero 2022].

5 ANEXOS

Los anexos adjuntados como parte complementaria al desarrollo del presente Trabajo de Integración Curricular se observan a continuación:

ANEXO I. Glosario

ANEXO II. Resultados Encuesta *Google Forms*

ANEXO III. Guía de usuario del *add-on*

ANEXO IV. Código del *add-on*