



Universidad Austral de Chile

Facultad de Ciencias de la Ingeniería
Escuela de Ingeniería Civil en Informática

PROTOTIPO DE PLATAFORMA WEB 2.0 PARA LA GEORREFERENCIACIÓN MUSICAL

Proyecto para optar al título de
Ingeniero Civil en Informática

PROFESOR PATROCINANTE
HENRIQUE CHEVREUX OLIVEIRA COELHO DIAS
BACHAREL EM CIÊNCIA DA COMPUTAÇÃO

PROFESOR CO-PATROCINANTE
TEODORO DANNEMANN ALVAREZ
INGENIERO INDUSTRIAL AMBIENTAL
MAGÍSTER EN FÍSICA DE SISTEMAS COMPLEJOS

PROFESOR INFORMANTE
LUIS ÁLVAREZ GONZÁLEZ
INGENIERO CIVIL ELECTRICISTA
MAGÍSTER EN INGENIERÍA INFORMÁTICA

CAMILO ALEXANDER ALARCÓN ROMERO

VALDIVIA – CHILE
2020

AGRADECIMIENTOS

Quiero agradecer enormemente a mi familia por el apoyo y gran sacrificio para que yo pueda ser un profesional, ellos son y serán siempre el pilar de mi vida.

Agradezco a mi madre, Sonia, por enseñarme la importancia del orden y del respeto en todos los ámbitos de la vida. Agradezco a mi padre, Juan, por demostrarme lo importante que es la dedicación y la responsabilidad. Agradezco a mi hermano menor, Guillermo, por su apoyo en crear y descubrir, siempre con su buena onda y sensibilidad en aspectos creativos.

También quiero agradecer de forma especial a mis dos abuelitos, a pesar de que tuvieron que partir en la mitad de este camino, siempre los querré...

Ha sido un largo camino, de mucho esfuerzo, con momentos altos y bajos, con penas, pero también con triunfos, tanto académicos como personales. Puedo considerar como un triunfo las grandes amistades formadas en estos años, sin duda, todos los viajes, las experiencias y momentos juntos los recordaré y atesoraré. Amistades que hicieron este paso por la universidad algo increíblemente grato.

Quiero agradecer también a todos los profesores y funcionarios que nos entregaron su conocimiento y sabiduría, y me guiaron en este camino.

A mi patrocinante, profesor Henrique, infinitas gracias por su tiempo y apoyo incondicional. Por esas conversaciones por horas, charlando del proyecto, la vida y de las proyecciones.

En fin, gracias a todos los que en algún sentido fueron parte de este extenso y complejo proceso.

ÍNDICE

ÍNDICE DE TABLAS.....	V
ÍNDICE DE FIGURAS.....	VI
RESUMEN.....	VIII
ABSTRACT	IX
1. INTRODUCCIÓN	1
1.1. Motivación	1
1.2. Impacto	2
1.3. Objetivos	3
1.3.1. Objetivo general	3
1.3.2. Objetivos específicos	3
2. CONTEXTO.....	4
2.1. Web 2.0.....	4
2.2. Streaming de música	4
2.3. Hábitos de escucha musical.....	6
2.3.1. Globalmente	6
2.3.2. En Chile.....	7
2.4. Revisión de mercado	7
2.4.1. Montevideo Sonoro	8
2.4.2. Pablo Morales & L@s Inmorales - Puntos Sobre el Plano	9
2.4.3. Nature Soundmap	10
2.4.4. Spotify Maps - Musical Cities.....	10
2.4.5. Google Maps	11
2.4.6. Wikipedia	12
2.4.7. Genius	13
2.5. Páginas web actuales de georreferenciación musical.....	14
2.5.1. Montevideo Sonoro	14
2.5.2. Puntos Sobre el Plano	16
3. DISEÑO	18
3.1. Requerimientos de diseño.....	18
3.1.1. Arquetipos de usuario	19
3.1.2. Perfil de comportamiento de usuario	20
3.1.3. Requerimientos funcionales	21
3.1.4. Requerimientos no funcionales	26
3.2. Diseño de interfaz.....	28
3.3. Diseño del software	29

3.3.1.	MEAN Stack	30
3.3.2.	MongoDB.....	31
3.3.3.	Express	33
3.3.4.	Angular.....	34
3.3.5.	NodeJS	37
3.3.6.	NPM	38
3.4.	Integración Continua	38
3.4.1.	Prácticas esenciales de Integración Continua	39
3.4.2.	Jenkins.....	39
3.4.3.	Bitbucket	40
3.4.4.	Compodoc	41
3.5.	Otras tecnologías utilizadas.....	41
3.5.1.	Spotify	41
3.5.2.	YouTube.....	42
3.5.3.	Google Maps	43
3.5.4.	Google Analytics.....	44
3.5.5.	UptimeRobot	45
3.5.6.	Disqus.....	46
3.5.7.	Auth0.....	47
3.6.	Diseño de navegación	48
3.6.1.	Ventana principal MusycGo	48
3.6.2.	Mapa principal	49
3.6.3.	Ranking de canciones cercanas	52
3.6.4.	Tendencias.....	53
3.6.5.	Canciones visibles.....	54
3.6.6.	Configuraciones	55
3.6.7.	Ingreso de canción.....	56
3.6.8.	Buscador general	57
3.6.9.	Estadísticas generales.....	58
3.6.10.	Comentarios de MusycGo.....	59
3.6.11.	Inicio de sesión.....	59
3.6.12.	Perfil de usuario	60
3.6.13.	Perfil de lugar de la canción.....	62
3.6.14.	Perfil maestro de canción	65
4.	IMPLEMENTACIÓN	67
4.1.	Metodología de trabajo.....	67
4.2.	Diagnóstico y despliegue	70
4.2.1.	Prueba de concepto	70
4.2.2.	Carpetas.....	70
4.2.3.	Despliegue.....	72
4.3.	Datos	72
4.3.1.	Modelo	72
4.3.2.	Respaldo.....	75

4.3.3. Petición de datos	76
4.4. Componentes de MusycGo	77
4.4.1. Mapa.....	78
4.4.2. Estadística	80
4.4.3. Ingreso de canción.....	80
4.4.4. Autenticación	81
4.4.5. Notificación.....	82
4.4.6. Buscador.....	83
4.4.7. Sitio 404	83
4.4.8. Perfil de usuario	83
4.4.9. Perfil maestro de la canción	84
4.4.10. Perfil del lugar de la canción.....	85
4.5. Otros recursos.....	86
4.5.1. Multi-idioma	86
4.5.2. Estilos de mapa	87
4.5.3. Base de datos de países	87
4.5.4. Servicio de Spotify	88
4.5.5. Geolocalización.....	89
4.6. Revisión de la versión lograda	90
5. VALIDACIÓN Y RESULTADOS.....	92
5.1. Nombre de la plataforma.....	92
5.2. Test de usabilidad.....	92
5.2.1. Elección del grupo.....	93
5.2.2. Desarrollo.....	93
5.2.3. Resultados	95
5.2.4. Consecuencia de las pruebas de usabilidad.....	100
6. CONCLUSIONES.....	101
6.1. Cumplimiento de objetivos.....	101
6.1.1. Revisar aplicaciones similares para realizar una interfaz interactiva.....	101
6.1.2. Investigar metodología y arquitectura adecuada para este proyecto.....	102
6.1.3. Priorizar funcionalidades para realizar un prototipo en el menor tiempo posible	102
6.1.4. Validar la aplicación a través de pruebas y feedback de los usuarios.....	102
6.1.5. Asegurar el tiempo requerido para el desarrollo de este proyecto, a través de la planificación y la estimación.....	102
6.1.6. Realizar diseño de la aplicación	102
6.1.7. Difundir la aplicación para impactar a más usuarios	102
7. TRABAJOS FUTUROS	103
8. REFERENCIAS	104
ANEXOS	106

Anexo A: Cuestionario aplicado a los usuarios en prueba de usabilidad	106
Anexo B: Resultados de la prueba SUS para cada pregunta por usuario	109
Anexo C: Vista versión móvil – responsive.....	110
Anexo D: Figuras de MusycGo	112

ÍNDICE DE TABLAS

Tabla	Página
Tabla 1. Incremento uso de streaming de música	7
Tabla 2. Descripción de categorías del método MoSCoW	19
Tabla 3. Necesidades de los usuarios	19
Tabla 4. Arquetipos de usuario	20
Tabla 5. Requerimientos funcionales para el módulo: Mapa Principal	22
Tabla 6. Requerimientos funcionales para el módulo: Ingreso de Canciones	24
Tabla 7. Requerimientos funcionales para el módulo: Buscador General	24
Tabla 8. Requerimientos funcionales para el módulo: Comentarios	25
Tabla 9. Requerimientos funcionales para el módulo: Estadísticas	25
Tabla 10. Requerimientos funcionales para el módulo: Perfil Maestro de Canciones	25
Tabla 11. Requerimientos funcionales para el módulo: Perfil de Usuario	25
Tabla 12. Requerimientos funcionales para el módulo: Perfil de Lugar de Canción	26
Tabla 13. Requerimientos funcionales para el módulo: Notificaciones	26
Tabla 14. Requerimientos no funcionales de seguridad.....	27
Tabla 15. Requerimientos no funcionales de usabilidad.....	27
Tabla 16. Requerimientos no funcionales de fiabilidad.....	27
Tabla 17. Requerimientos no funcionales de rendimiento.....	28
Tabla 18. Colección de música en la base de datos	73
Tabla 19. Colección de usuarios en la base de datos	75
Tabla 20. Puntaje y puntuación SUS.....	99
Tabla 21. Grado para puntajes SUS	100

ÍNDICE DE FIGURAS

Figura	Página
Figura 1. Vista principal Montevideo Sonoro.....	8
Figura 2. Vista principal Puntos Sobre el Plano	9
Figura 3. Vista principal Nature Soundmap.....	10
Figura 4. Vista principal Spotify Maps - Musical Cities	11
Figura 5. Vista principal Google Maps	11
Figura 6. Vista de la portada de Wikipedia.....	12
Figura 7. Categoría "Canciones sobre ciudades" en Wikipedia.....	13
Figura 8. Subcategoría "Canciones sobre ciudades de Chile" en Wikipedia	13
Figura 9. Ejemplo de perfil de canción en Genius	14
Figura 10. Información general Montevideo Sonoro	15
Figura 11. Información del marcador seleccionado en sitio Montevideo Sonoro	16
Figura 12. Marcadores en el mapa web de Puntos Sobre el Plano	17
Figura 13. Información del marcador seleccionado en Puntos Sobre el Plano	17
Figura 14. Vista principal prueba de concepto.....	29
Figura 15. Recomendaciones canciones cercanas en prueba de concepto.....	29
Figura 16. Diagrama de tecnologías de MusycGo	30
Figura 17. Concepto MEAN Stack	31
Figura 18. <i>Ranking DBMS</i>	32
Figura 19. DB-Engines <i>ranking</i> de base de datos orientadas a documentos	33
Figura 20. Información de la base de datos de MusycGo	33
Figura 21. Rutas para peticiones a la base de datos de música vía Express	34
Figura 22. Ejemplo de codificación en Angular	35
Figura 23. Monitoreo de la instancia de NodeJS vía PM2.....	38
Figura 24. Servidor de Integración Continua para MusycGo vía Jenkins	40
Figura 25. Versionamiento de código MusycGo vía Bitbucket	40
Figura 26. Documentación de código fuente de MusycGo vía Compodoc	41
Figura 27. Ejemplo de <i>iframe</i> de Spotify en MusycGo	42
Figura 28. Ejemplo de <i>iframe</i> de YouTube en MusycGo	43
Figura 29. Ejemplo de uso del mapa Google Maps para el ingreso de una canción.....	44
Figura 30. Algunas estadísticas de MusycGo vía Google Analytics	45
Figura 31. Monitorización del estado de MusycGo vía UptimeRobot	46
Figura 32. Ejemplo de uso de Disqus para el ingreso de comentarios en MusycGo	47
Figura 33. Registro e inicio de sesión en MusycGo vía Auth0.....	47
Figura 34. Diagrama de navegación MusycGo.....	48
Figura 35. Ventana principal MusycGo	49
Figura 36. Ventana mapa principal de MusycGo.....	50
Figura 37. Marcador mapa principal e información extra.....	51
Figura 38. Lista de canciones cercanas y por país	52
Figura 39. <i>Ranking</i> de canciones cercanas, radio de alcance e información extra	53
Figura 40. Tendencias en reproducciones y visualizaciones.	54

Figura 41. Canciones visibles e información extra	55
Figura 42. Configuración de mapa, filtros y cambio de estilo	55
Figura 43. Ventana ingreso de canción	56
Figura 44. Ventana de buscador general de colaboraciones	57
Figura 45. Ventana de estadísticas generales	58
Figura 46. Ventana de comentarios de MusycGo	59
Figura 47. Ventana de registro e inicio de sesión	59
Figura 48. Información del usuario autenticado	60
Figura 49. Ventana del perfil de usuario	61
Figura 50. Ventana del perfil del lugar de la canción, parte 1: Encabezado	63
Figura 51. Ventana del perfil del lugar de la canción, parte 2: Cuerpo	64
Figura 52. Ventana del perfil del lugar de la canción, parte 3: Cuerpo	65
Figura 53. Ventana del perfil maestro de la canción	66
Figura 54. Metodología Scrum	67
Figura 55. Administración de proyecto MusycGo vía Trello	69
Figura 56. Estructura de carpetas proyecto MusycGo	70
Figura 57. Respuesta de petición de datos a API MusycGo	77
Figura 58. Notificaciones de cambios en colaboraciones	82
Figura 59. Ventana de URL inválida	83
Figura 60. Perfil público de usuario	84
Figura 61. Formato respuesta SUS	94
Figura 62. Histograma cantidad de usuarios en un rango etario	95
Figura 63. Cantidad de usuarios que usaron MusycGo por un rango de tiempo	96
Figura 64. Porcentaje de usuarios según perfil	96
Figura 65. Porcentaje de ingresos a MusycGo por navegador	97
Figura 66. Porcentaje de ingresos a MusycGo por dispositivo	97

RESUMEN

Muchos son los lugares de la Tierra sobre el cual el ser humano ha escrito alguna canción, pero hay mucha gente que no sabe la existencia de estas. A medida que se va modernizando la sociedad, se van perdiendo las raíces de los pueblos que la componen, aquellas anécdotas y momentos que son bien descritas a través de la música.

MusycGo pretende rescatar, fomentar y potenciar el patrimonio musical de todas las ciudades, pueblos y lugares del mundo, ofreciendo una plataforma digital interactiva que permita crear una relación entre la música y nuestras ciudades, pueblos, calles, monumentos, ríos, etc.

Las investigaciones llevadas a cabo muestran el increíble impacto y repercusión que ha generado en los últimos años la disponibilidad de la música en medios digitales de *streaming*. Estas mismas plataformas incentivan a artistas a mostrarse al mundo de una manera más directa, rápida y efectiva.

Los resultados logrados definen una metodología y arquitectura que abastecen las características de una aplicación de exploración y contribución musical. Se analizaron los comportamientos de arquetipos de usuarios, los cuales permitieron listar las funcionalidades requeridas como historias de usuario, indicando su nivel de prioridad. Luego, para el desarrollo, se tuvieron los artefactos de diseño y para validar las funcionalidades se aplicaron pruebas de usabilidad a usuarios reales, permitiendo el *feedback*.

Actualmente existen plataformas web de terceros que exhiben algunas de las características de este proyecto, pero cada una de ellas de forma aislada. La principal ventaja es que se desarrolló una plataforma basada en Web 2.0, es decir, permite la colaboración de los usuarios, pudiendo retroalimentar, calificar y actualizar la información que se mostrará. También a diferencia de otras aplicaciones, este proyecto tiene la cualidad de poder ver la música que te rodea según tu ubicación.

ABSTRACT

There are many places on Earth on which the human being has written a song, but there are many people who don't know the existence of these. As society modernizes, the roots of the peoples that compose it are lost, those anecdotes and moments that are well described through music.

MusycGo aims to rescue, promote and enhance the musical heritage of all cities, towns and places in the world, offering an interactive digital platform that allows creating a relationship between music and our cities, towns, streets, monuments, rivers, etc.

The research carried out shows the incredible impact and repercussion that the availability of music in digital streaming media has generated in recent years. These same platforms encourage artists to show themselves to the world in a more direct, fast and effective way.

The results achieved define a methodology and architecture that supply the characteristics of an application for exploration and musical contribution. The behaviors of user archetypes were analyzed, which allowed listing the required functionalities as user stories, indicating their priority level. Then, for development, the design artifacts were taken and to validate the functionalities, usability tests were applied to real users, allowing feedback.

Currently there are third-party web platforms that exhibit some of the characteristics of this project, but each one in isolation. The main advantage is that a Web 2.0-based platform was developed, that is, it allows the collaboration of users, being able to provide feedback, qualify and update the information that will be displayed. Also unlike other applications, this project has the quality of being able to see the music around you according to your location.

1. INTRODUCCIÓN

Con el paso del tiempo se desarrollaron cientos de teorías para explicar el sentido de la música, lo que nadie puede explicar con certeza es que tienen los sonidos que pueden tocarnos el sistema nervioso y emocionarnos a puntos que ninguna otra cosa puede hacerlo. La música tiene por ende mucho de misterio, de magia, y presenta para nosotros un mundo que no somos totalmente capaces de comprender, pero al que llegamos una y otra vez de forma irremisible (Pérez Porto & Gardey, 2012).

A pesar de que la música es increíblemente íntima, permite a los artistas transmitir tanto directa como indirectamente un mensaje, anécdota o una emoción. Cuando son capaces de comunicar con palabras y melodías, es ahí cuando revelan su parte más profunda. En el momento que algunas personas comprenden las razones de una canción, ellos tienen un gran conocimiento que pueden transmitir a las demás personas, pero que puede llegar a ser complicado si no se cuenta con los medios.

La era digital, la inmensa cantidad de datos e información, la sociedad cada vez más interconectada y la innovación, prácticamente incentivan a generar y descubrir, si a esto se añaden los intereses y conocimientos de cada persona, puede conllevar a una inteligencia colectiva muy potente.

El diseño y confección de una plataforma que permita aprovechar la inteligencia colectiva en aspectos culturales es un gran apoyo para aquellos que la quieren preservar y difundir. Cuando la música es considerada el arte cultural que tienen los: barrios, pueblos, ciudades, países e incluso los continentes, es indiscutiblemente algo que se debe proteger y comunicar.

Hoy en día es común portar un dispositivo GPS¹ encima, ya que se utiliza para llegar a sitios, saber dónde estamos, o inclusive para buscar curiosos “animales virtuales”. La presencia en dispositivos como computadores personales y *smartphones* ofrecen ventajas para aprovechar los datos del entorno, para poder recomendar y obtener resultados personalizados acordes a la geolocalización.

Una canción es más fácil sentirla y reproducirla que entenderla o definirla. Todos tenemos nociones de qué es la música, pero ¿cuántos pueden poner en palabras cuáles son las cosas que le dan sentido e identidad?

1.1. Motivación

“La música es una de las expresiones creativas más íntimas del ser, ya que forma parte del quehacer cotidiano de cualquier grupo humano tanto por su goce estético como por su carácter funcional y social. La música nos identifica como seres, como grupos y como

¹ GPS de siglas en inglés *Global Positioning System* (Sistema de Posicionamiento Global).

cultura, tanto por las raíces identitarias como por la locación geográfica y épocas históricas. Es un aspecto de la humanidad innegable e irremplazable que nos determina como tal” (Angel, Camus, & Mansilla, 2008).

La motivación para llevar a cabo este proyecto nace como idea de la empresa Puentes Digitales². Quienes notaron que existía un interés por parte de muchas personas de conocer la pertenencia de la música con ciertos lugares, y hay otras personas que tienen el conocimiento de este tipo de información pero que no saben cómo divulgarlo.

A medida que se va modernizando la sociedad, se van perdiendo las raíces de los pueblos que la componen, aquellas anécdotas y momentos de antaño que son bien descritas a través de música. MusycGo³ (nombre del presente proyecto desarrollado, visitar sección 5.1 para más información) permite almacenar esta información, para que sean recordadas y distinguidas por aquellos que quieren saber más sobre el lugar en donde están, enriqueciendo al usuario y proveyéndoles de la cultura de donde estén.

El objetivo del proyecto MusycGo es recopilar el patrimonio musical de las ciudades, pueblos y lugares del mundo, para luego divulgar esta información georreferenciada de manera interactiva e intuitiva. Por lo tanto, el proyecto tiene la particularidad de enriquecer el patrimonio cultural de una localidad, además de satisfacer la necesidad de información instantánea que tienen las personas en los tiempos actuales. De acuerdo con esto último, según los resultados de (Kemp, 2020), hay 5.19 mil millones de usuarios de dispositivos móviles y 4.54 mil millones son usuarios activos de Internet en el mundo, además el usuario promedio de Internet pasa seis horas y 43 minutos en línea cada día. Esto nos muestra un ambiente propicio para el desarrollo de MusycGo.

La oportunidad descrita ocurre debido a la no existencia de algún instrumento que busque integrar la información musical digital en uno, y que sea específicamente para enfatizar alguna localidad.

MusycGo se abre a la escena musical en el sentido de que puede potenciar y motivar a artistas emergentes para darse a conocer, incluyendo georreferenciaciones de sus trabajos musicales y las letras que expresan vivencias o momentos que son inspiración para los artistas. Cada artista, independiente de su trayectoria tiene la posibilidad de pertenecer a la base de datos de MusycGo y así complementar el patrimonio cultural-musical.

1.2. Impacto

La información preparada por (GfK, 2018) sobre “Chile Digital 2018” (caso de estudio con tamaño de 4.800 personas) muestra que a la mayoría de los chilenos les encanta la tecnología (60%), además los chilenos mencionaron un gran interés por la música (84%),

² <http://puentesdigitales.cl/>

³ <https://www.musycgo.com/>

lo que demuestra que la sinergia entre los conceptos: música y tecnología (ambos forman los pilares de MusycGo), puede conllevar a un alto impacto. Además, hay que señalar que la música tuvo el mayor porcentaje de interés con respecto a otros temas, entre los que se encontraba: protección animal, medio ambiente, viajes, bienestar, contingencia, cocina, cine, tecnología, moda y deportes en general.

Según la “Encuesta Música Chilena 2018” (AAM, 2018), al 64% de los chilenos le gusta la música nacional, mientras que el 51% declara que escucha bastante música hecha por artistas locales. Con respecto a si las personas les gusta la música hecha por artistas locales, un 64% de los encuestados indicó que le agrada mucho, lo que representa un aumento significativo con respecto al 54% del año 2016. Por otra parte, quienes no escuchan música hecha por artistas locales (49%), atribuyen esto a su poca difusión radial y a que no saben dónde encontrarla.

Con lo anterior, se puede inferir que a la gente le gusta cada vez más la música de chilenos, y podría aumentar si las personas conocieran una plataforma donde encontrarla. El impacto social repercute en que las personas pueden conocer sus raíces musicales, aprendiendo de lo que los rodea, además, si más personas escuchan música de artistas locales, están aportando en la divulgación de la música, permitiendo generar mayores ingresos a estos artistas.

Las personas podrán aumentar su conocimiento cultural, además de poder acceder fácilmente a la música de una localidad y también a la que los rodean, permitiendo el incremento de las relaciones entre los artistas, su música, las personas y la localidad.

1.3. Objetivos

1.3.1. Objetivo general

Fomentar la cultura musical regional, a través de una plataforma digital interactiva que permita crear una relación entre la localidad y la música.

1.3.2. Objetivos específicos

1. Revisar aplicaciones similares para realizar una interfaz interactiva.
2. Investigar metodología y arquitectura adecuada para este proyecto.
3. Priorizar funcionalidades para realizar un prototipo en el menor tiempo posible.
4. Validar la aplicación a través de pruebas y *feedback* de los usuarios.
5. Asegurar el tiempo requerido para el desarrollo de este proyecto, a través de la planificación y la estimación.
6. Realizar diseño de la aplicación.
7. Difundir la aplicación para impactar a más usuarios.

2. CONTEXTO

2.1. Web 2.0

La “*World Wide Web*”, conocida como la web (sistema de documentos de hipertexto vinculados, accesibles por Internet), nació a principios de la década de 1990 y en sus inicios solo ofreció contenido en formato textual agrupados en hipervínculos o *links*. En la actualidad los sitios web ya no son solo textos e hipervínculos, el contenido y la interacción con los usuarios ha cambiado, por eso apareció la “nueva web”, la Web 2.0.

Se le considera Web 2.0 a la evolución de la manera en que se comunica la información, en cómo pasamos de un conjunto de plataformas estáticas de comunicación y colaboración unidireccional, a una web con contenido dinámico, que permite a los usuarios colaborar e interactuar entre ellos, generando contenido de manera multidireccional. En la “Web 1.0”, la información no era actualizada frecuentemente (tarea a cargo de los programadores). Con la evolución los usuarios dejan de ser pasivos dentro de la web, ya que participan y contribuyen en el contenido de los sitios, siendo capaces de crear, dar soporte, compartir, y formar parte de una comunidad que puede crecer a proporciones a nivel global.

El origen del término Web 2.0, fue acuñado por Tim O'Reilly fundador de la empresa O'Reilly Media (empresa que se dedica a la editorial de libros de tecnología) en el año 2003 y su primer uso en público fue en una conferencia en octubre del año 2004.

Otro punto importante es que las personas sin siquiera saber sobre programación pueden generar contribuciones y conocimiento, con contenido muy variado, como: textos, imágenes, audio, videos, entre otros. Todo esto facilita el acercamiento e interacción de las personas con una web dinámica en donde puedan generar y valorar la información, que en la era actual es algo muy relevante.

Algunos ejemplos de tecnologías y herramientas Web 2.0 son: foros, redes sociales, *microblogging*, *wiki*, *blogs*, noticias, mensajería instantánea, etc. En estos ejemplos la Web 2.0 permite: instruir, investigar, publicar, informar, notificar, denunciar, promocionar productos y servicios, entre otros.

En síntesis, la Web 2.0 es un concepto “nuevo” que revoluciona la Internet, ya que la convierte en una plataforma que tiene la capacidad de estar en constante cambio, una permanente construcción, y mantiene un interés creciente, debido a que los contenidos pueden ser creados, modificados y compartidos por los usuarios.

2.2. Streaming de música

Hace más de una década, todas las personas que querían escuchar su música personal en cualquier lugar tenían que portar diferentes dispositivos de reproducción y realizar tediosos procesos de transferencias de archivos, y en algunos casos liberar espacio en el almacenamiento de estos dispositivos para poder tener toda su música favorita. Como solución, algunas empresas crearon sus propias plataformas de música en *streaming*, con

esto es posible escuchar todas tus canciones favoritas en cualquier parte y a todas horas, dejando de lado la música “física” en pro de inmensas bibliotecas de música *online*, en donde cada plataforma ofrece sus propias características, funcionalidades y número de canciones. Por otra parte, tu biblioteca de música online puedes tenerla en cualquier dispositivo que sea compatible con la plataforma (web, móvil y multiplataforma), por lo que puedes escucharlas en todo momento y lugar.

Según la fuente (Bertolini, Cabrera, & Ovalle, 2019) “Las plataformas de *streaming* muestran de cierta forma cómo va cambiando el consumo en la sociedad digital actual, cómo se ha ido transformando la audiencia y finalmente el reflejo de una tendencia a nivel global de acceso a nuevas tecnologías y plataformas denominadas de *streaming*”.

Algunas plataformas que actualmente están disponible son: Spotify, Last.fm, Deezer, Movistar Música, SoundCloud, Amazon Prime Music, Google Play Music, Apple Music y YouTube Music.

Debido a que en este proyecto se implementan funcionalidades ligadas a la plataforma de Spotify, se muestran algunas cifras, el impacto que generó su aparición y el furor en la actualidad.

Spotify es un servicio de música, podcasts y videos digitales en *streaming* que cuenta con millones de canciones y otros contenidos de artistas de todo el mundo. Actualmente es una de las plataformas más populares de transmisión de música en general, ya que lidera la participación en el mercado global con el 35,6% (Mulligan, 2019).

A nivel global Spotify ha provocado una revolución en la industria musical y en específico en el consumo de la música, la compañía sueca cuenta con más de 35 millones de canciones en su catálogo. Según (El País, 2018) “es el servicio que más retorno económico proporciona hoy por hoy al negocio de la música (9.2 mil millones de dólares a 31 de diciembre de 2017)”.

Además, en el presente proyecto se tiene una base estable mediante la utilización y el impacto de las plataformas de *streaming* del último tiempo, de hecho, el impacto que ha tenido Spotify dentro de la industria de la música en *streaming* es tan masivo que el presente año 2020 se realizó en México una ceremonia que corresponde a la primera entrega de premios musicales de Spotify, llamado “Spotify Awards”, cuyos ganadores fueron determinados por las reproducciones, patrones y hábitos de escucha de los usuarios en la plataforma de Spotify en el año 2019 (Infobae, 2020).

Según (Hernández Ruza, 2019) durante el tercer trimestre de 2019 el total de usuarios activos mensuales de Spotify creció en un 30% a 248 millones. La compañía superó los pronósticos de crecimiento de sus suscriptores *premium*, los cuales crecieron un 31% en comparación con el año anterior para alcanzar los 113 millones.

Las estadísticas dentro de fronteras chilenas son igualmente alentadora y favorable para Spotify, ya que según (Aguirre A., 2017) “52 millones de canciones son reproducidas

semanalmente, lo que equivale a 188 millones de minutos de *streaming* a la semana, y con 173 minutos como promedio de escucha por usuario”.

Según (IMI Chile, 2018) el informe emitido por la Industria Musical Independiente de Chile (IMI Chile) y La Sociedad de Derechos de Autor (SCD), afirma que “Spotify hoy juega un rol crucial en la internacionalización de la música independiente, existiendo un antes y un después”.

2.3. Hábitos de escucha musical

2.3.1. Globalmente

La mayoría de las personas afirman que son amantes o fanáticos de la música, esto es lo que señala La Federación Internacional de la Industria Fonográfica (IFPI), en el informe (IFPI, 2019).

Actualmente existe un aumento en el querer disfrutar de la música, esto debido a su variada disponibilidad global. El aumento en este hábito se ve en que los fanáticos están descubriendo géneros locales, para enriquecer sus vidas y adaptarse a su cultura.

IFPI es la organización que representa a la industria de la música grabada en todo el mundo. Esta organización lanzó en el mes de septiembre del año 2019 un informe realizado entre abril-mayo del mismo año, que examinó la forma en que los consumidores de música (entre 16 y 64 años de edad) interactúan con la música grabada en 21 países.

Dentro de la información obtenida se menciona que, existiendo diferentes maneras de acceder la música, el fácil acceso y la disponibilidad de masivas bibliotecas de música *online* hacen que el *streaming* sea la opción más popular (transversal en todas las edades).

Se señalan los hallazgos relevantes dentro de este informe del año 2019:

- El 89% escucha música a través de música de alguna plataforma de *streaming*.
- El uso de servicios de *streaming* tiene una tasa de crecimiento más alta en el grupo de edad de 35 a 64 años. En este rango de edad, el 54% usaron este tipo de servicios en el mes de agosto 2019, con un incremento de ocho puntos porcentuales desde el año 2018.
- El tiempo dedicado a escuchar música cada semana es de 18 horas en promedio (el año 2018 fueron 17.8 horas), es decir, más de 2.6 horas diarias, lo cual equivale a escuchar 52 canciones de tres minutos diariamente.
- La mayoría de los encuestados (54%) considera que ama o es fanático de la música. Entre 16 y 24 años, el porcentaje aumenta al 63%.

La Tabla 1 muestra el incremento del uso del *streaming* de música, lo que indica que es más notorio en las personas de mayor edad. Las razones de estos incrementos se deben a que el acceso es inmediato a millones de canciones (para el 62% de los encuestados),

pueden escuchar lo que quieren y cuando quieren (para el 61%) y también es la forma más conveniente de escuchar música (para el 41%).

Tabla 1. Incremento uso de streaming de música⁴

Grupo de edad	Uso de streaming de música 2018	Incremento desde 2018 a 2019 [p.p.]
Global	64%	+7
16-24	83%	+5
25-34	75%	+6
35-44	64%	+9
45-54	53%	+8
55-64	44%	+9

2.3.2. En Chile

Un estudio del equipo GfK Adimark y SCD (AAM, 2018) realizado entre junio-julio del año 2018, determinó que opinión tienen los chilenos de la música, cuya muestra representa a un número de 502 hombres y 500 mujeres mayores de 17 años, distribuidos en zona norte, centro y sur de Chile.

Dentro de los resultados obtenidos se enumera:

- El 84% de las personas declara tener interés en la música.
- El 45% de los encuestados escucha música al estar en Internet.
- El 64% de las personas siente un agrado por la música chilena.
- Con relación al año 2016, en el año 2018 aumenta el porcentaje de personas que escucha “muchísima” o “bastante” música, el aumento fue del 72% al 77% (declaran que la causa del incremento puede deberse a la consolidación de aplicaciones que facilitan el acceso a la música, como Spotify).
- Los rangos etarios que más disfrutan de la música son de 18 a 25 años (96%) y de 26 a 30 años (84%). Además, las personas de 56 o más años tuvieron un incremento muy importante con relación a este criterio, ya que entre 2016 y 2018 aumentaron de 47% a 67%, el cual corresponde al rango etario que obtuvo el mayor incremento.
- El 57% de los encuestados están de acuerdo en que la música chilena debería ser el 50% del contenido musical que se transmita en las radios.

2.4. Revisión de mercado

Para poder responder al primer objetivo específico, el cual alude a “Revisar aplicaciones similares para realizar una interfaz interactiva”, se realizó una revisión de la situación

⁴ <https://promocionmusical.es/ifpi-music-listening-2019>

actual para identificar funcionalidades y características que permitan llevar a cabo un proyecto intuitivo y también innovador.

A continuación, se definen los criterios de selección:

- Páginas y/o aplicaciones que si bien no hagan uso de Web 2.0, incorporen georreferenciación musical.
- Página y/o aplicación que haga uso de Web 2.0 para el ámbito musical.
- Mapas interactivos.

Como resultado de la revisión de mercado o situación actual con respecto al proyecto, podemos mencionar algunas aplicaciones y páginas web.

2.4.1. Montevideo Sonoro

Montevideo Sonoro⁵ es un proyecto colaborativo de georreferenciación y divulgación del patrimonio musical de la ciudad de Montevideo, Uruguay. Funciona en la web, donde se pueden encontrar aproximadamente trescientas canciones vinculadas a los barrios de la capital uruguaya (ver Figura 1).

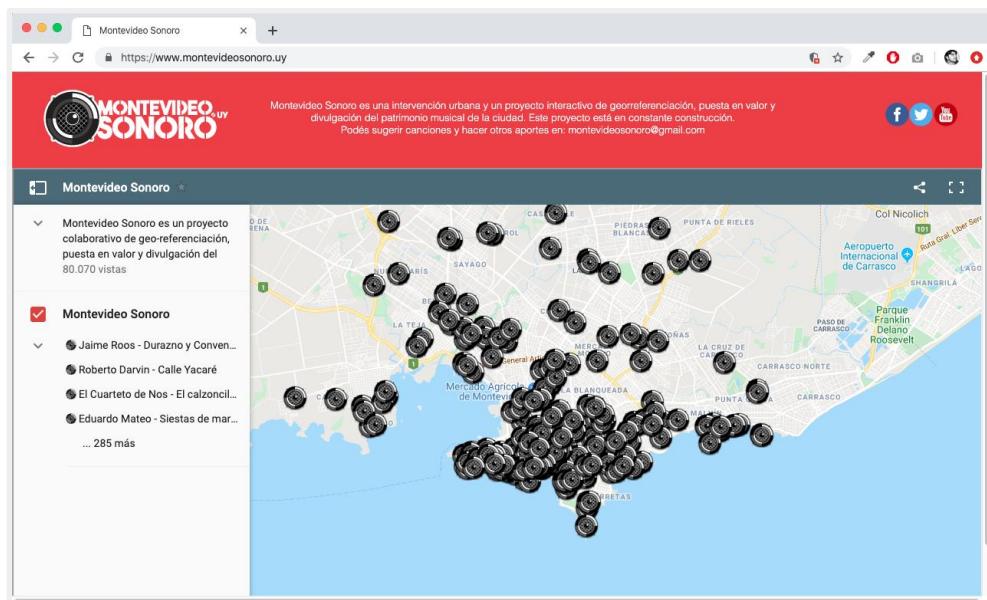


Figura 1. Vista principal Montevideo Sonoro

Montevideo Sonoro es también una intervención urbana, ya que los creadores del proyecto añaden códigos QR donde se encuentra una canción en las calles de Montevideo, por lo que las personas pueden escanear los códigos con sus celulares para acceder al contenido multimedia de la canción.

⁵ <http://www.montevideosonoro.uy/>

Por otra parte, existe un libro titulado “Montevideo Sonoro, La vuelta a la ciudad en 300 canciones”, el cual presenta lecturas acerca de las temáticas de las canciones y reflexiones sobre la motivación que impulsó la creación de cada una de ellas.

Algunas de las funcionalidades que presenta la versión web se asemejan bastante a las funcionalidades de este proyecto de título: música, reproductor, información en marcadores, la pantalla inicial es un mapa, entre otros. En sí es una aplicación muy completa, con gran potencial, pero aun así carece de una colaboración más directa por parte de usuarios, a través de publicaciones musicales desde la web.

2.4.2. Pablo Morales & L@s Inmorales - Puntos Sobre el Plano

Esta aplicación web nace en el año 2017 como idea de una banda chilena que quiso difundir su trabajo mediante un sitio web interactivo.

Puntos Sobre el Plano⁶ es una aplicación netamente dedicada a la música y ubicación geográfica del arte musical de la banda Pablo Morales & L@s Inmorales, de modo tal que permite relacionar canciones y lugares. También cabe mencionar que incorpora funcionalidades relevantes como un reproductor para poder escuchar las canciones del artista, y la posibilidad de ver información extra de los marcadores en el mapa (ver Figura 2).

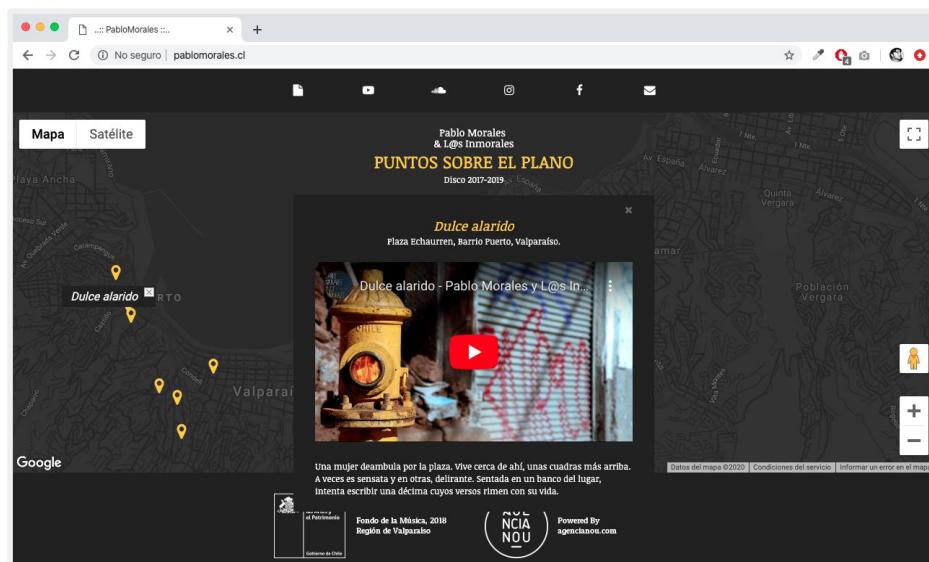


Figura 2. Vista principal Puntos Sobre el Plano

⁶ <http://pablomorales.cl/>

2.4.3. Nature Soundmap

Nature Soundmap⁷ es una aplicación web que reúne las contribuciones de un grupo de grabadores profesionales de la naturaleza de todo el mundo. Estas personas han colaborado para desarrollar una forma divertida e interactiva de explorar los sonidos naturales del planeta. Usando grabaciones de campo de alta calidad, el proyecto reúne algunos de los sonidos más interesantes e inspiradores de la naturaleza.

Destaca la parte visual, interactiva y amigable. También se agrega un reproductor de sonido en su ventana emergente al presionar un marcador (ver Figura 3).

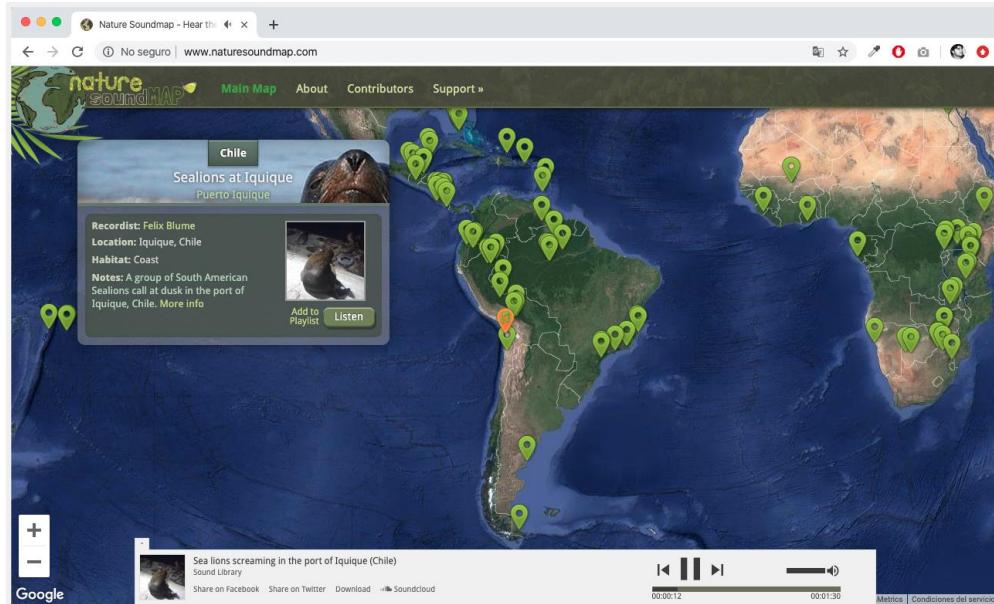


Figura 3. Vista principal Nature Soundmap

2.4.4. Spotify Maps - Musical Cities

Desde Spotify quisieron mostrar las canciones más escuchadas en todo el mundo a través de un mapa interactivo llamado Spotify Maps - Musical Cities⁸, en donde con un clic se puede interactuar con lo más escuchado de ese lugar.

Destaca el uso de un mapa para mostrar una lista de reproducción según una región específica del planeta (ver Figura 4). Otro punto importante es que hace uso de su propia API⁹ de desarrollo, tal cual se utilizó en este proyecto de título.

⁷ <http://www.naturesoundmap.com/>

⁸ <https://spotifymaps.github.io/musicalcities/>

⁹ API de siglas en inglés *Application Programming Interface* (Interfaz de Programación de Aplicaciones).

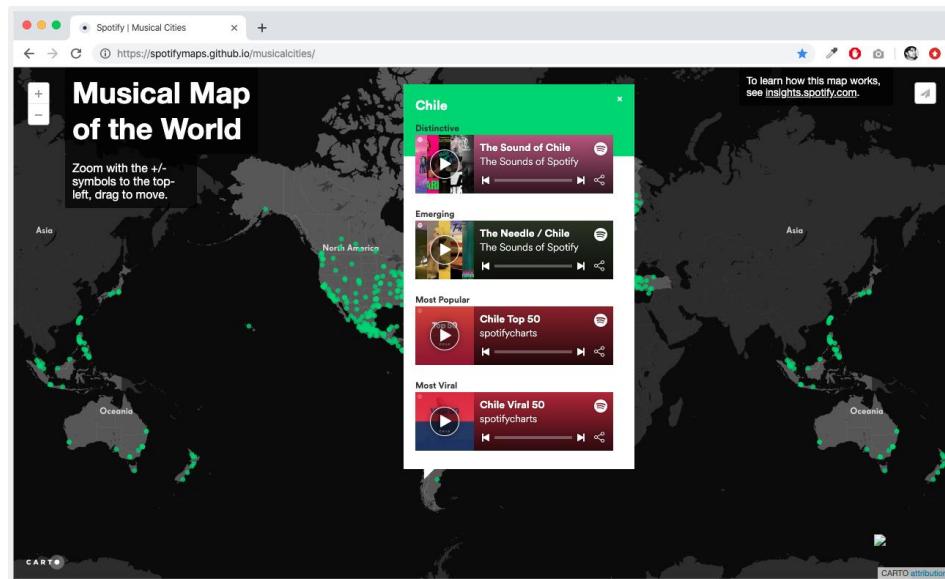


Figura 4. Vista principal Spotify Maps - Musical Cities

2.4.5. Google Maps

Google Maps¹⁰ es una plataforma web que permite explorar mapas, encontrar puntos de interés y obtener direcciones de cualquier lugar del mundo (ver Figura 5). Una de sus características es que ofrece tres opciones de diseño: normal, satélite y terreno; en función de las necesidades y preferencias de cada usuario. Además, Google Maps permite añadir nuevos destinos a una ruta con un sólo clic.

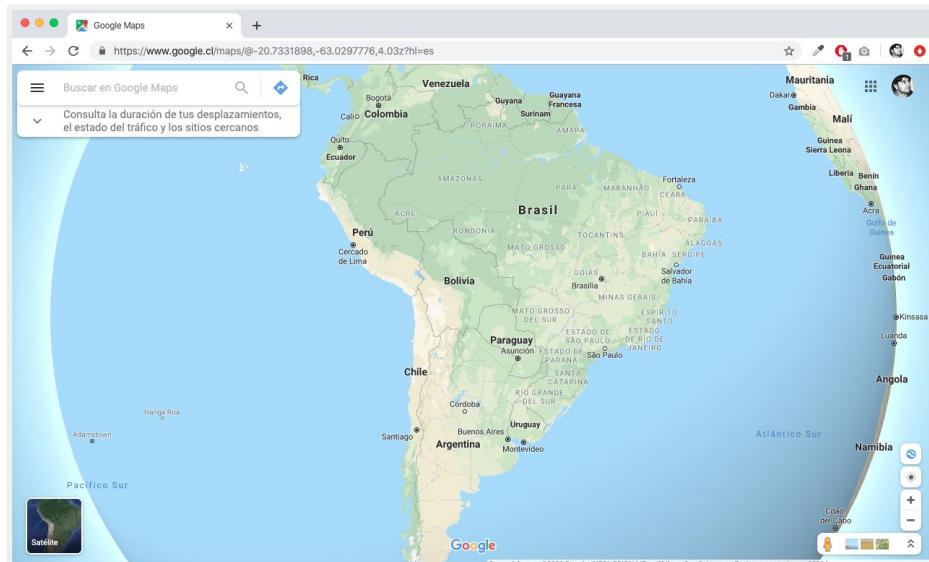


Figura 5. Vista principal Google Maps

¹⁰ <https://www.google.cl/maps/>

Es una plataforma referente en georreferenciación (restaurantes, hoteles, bares, cafeterías, etc.).

Google Maps tiene un alto grado de interactividad, lo que lo hace una aplicación fácil de usar, intuitiva y confiable.

Por otra parte, desde el año 2005 Google decide alterar su idea de servidor de aplicaciones de mapas en la web, ofreciendo una API, dándole la posibilidad a los usuarios de desarrollar aplicaciones multiplataforma. Este proyecto utilizada funcionalidades de dicha API y variados elementos que provee Google Maps para confeccionar los mapas y darles ciertas características.

2.4.6. Wikipedia

Wikipedia¹¹ es una enciclopedia libre, políglota y editada de manera colaborativa (ver Figura 6). Es administrada por una organización sin ánimo de lucro cuya financiación está basada en donaciones. Sus más de 50 millones de artículos en 300 idiomas han sido redactados en conjunto por voluntarios de todo el mundo, además permite que cualquier persona pueda sumarse al proyecto para editarlos.



Figura 6. Vista de la portada de Wikipedia

Se destaca la posibilidad colaborar con información, es decir, hay un interés en compartir contenido de forma colaborativa. Por otro lado, la música está presente, en particular debido a la existencia de listas que muestran las canciones que “hablan” sobre ciertas

¹¹ <https://es.wikipedia.org/>

ciudades del mundo (ver Figura 7) y en específico, se muestran algunas canciones en Chile (ver Figura 8).

The screenshot shows a standard Wikipedia category page titled "Categoría:Canciones sobre ciudades". The page includes a sidebar with various links like "Portada", "Herramientas", and language options. The main content area lists subcategories under letters B through N, each with a list of specific song titles. Below this is a section for "Páginas en la categoría «Canciones sobre ciudades»" which lists 23 pages, also organized by letter (B through M). At the bottom, there's a footer with categories like "Canciones sobre regiones" and "Localidades".

Figura 7. Categoría "Canciones sobre ciudades" en Wikipedia

This screenshot shows a subcategory page titled "Categoría:Canciones sobre ciudades de Chile". It features a sidebar and a single subcategory "S" which points to "Canciones sobre Santiago de Chile" (3 págs.). Below this is a section for "Páginas en la categoría «Canciones sobre ciudades de Chile»" which lists 3 pages, also organized by letter (L and P). The footer includes categories like "Canciones sobre ciudades" and "Canciones sobre Chile".

Figura 8. Subcategoría "Canciones sobre ciudades de Chile" en Wikipedia¹²

2.4.7. Genius

Finalmente tenemos la aplicación web, Genius¹³, fundada en 2009, cuya finalidad es ayudar a comprender el significado de canciones, esto a través de la explicación del significado y qué pensaba el artista en el momento de componer la letra.

¹² https://es.wikipedia.org/wiki/Categoría:Canciones_sobre_ciudades

¹³ <https://genius.com>

Lo positivo de Genius es que permite la colaboración a través de la inserción de comentarios, opiniones, suposiciones o citas de entrevistas con los músicos creadores. Su interacción permite visualizar la letra de la canción, subrayando fragmentos para poder insertar comentarios, en donde los demás usuarios pueden valorar dicha información (ver Figura 9).

The screenshot shows a web browser displaying the Genius website for the song "Here Comes the Sun" by The Beatles. At the top, there's a navigation bar with links for FEATURED, CHARTS, VIDEOS, COMMUNITY, SHOP, and social media icons. Below the navigation is a large image of the Beatles crossing Abbey Road. The main content area has a yellow header with the song title and artist name. The lyrics are listed in sections: [Intro], [Verse 1], [Chorus], and [Verse 2]. To the right of the lyrics, there's a sidebar with information about the song's history, including a quote from Harrison. Below the lyrics, there's a player bar with playback controls and a note about Eric Clapton's house.

Figura 9. Ejemplo de perfil de canción en Genius

2.5. Páginas web actuales de georreferenciación musical

Hay que destacar dos referencias importantes recolectadas en la revisión de mercado, debido a que la idea general, el conjunto de funcionalidades y características, contemplan un acercamiento relevante al proyecto actual, además se pone en evidencia el gran interés internacional por conocer y preservar ámbitos más allá que la propia música.

Las dos referencias siguientes fueron descritas en las secciones 2.4.1 y 2.4.2 respectivamente. A continuación, se detallan aspectos sobre sus funcionalidades.

2.5.1. Montevideo Sonoro

Como se mencionó en la sección 2.4.1, Montevideo Sonoro es un proyecto que divulga el patrimonio musical de la capital uruguaya.

Cuando se hace ingreso a la ventana principal (ver Figura 1) nos encontramos con un mapa creado con la aplicación “Google My Maps”, y sobre este tenemos una cabecera que muestra el logotipo, información para contextualizar la aplicación y los enlaces a redes sociales.

Al ser un mapa creado con Google My Maps, funciona con Google Drive. Hay que mencionar que este mapa según la configuración que propuso el creador no permite la colaboración directa en el mapa. Si bien la opción puede ser activada, en esta plataforma no lo está. La manera en que las personas pueden ayudar a poblar la aplicación es enviando un correo a los creadores con la información correspondiente y luego podrán ser incluidas. Por otra parte, este tipo de mapa se puede incluir fácilmente en un sitio web externo, agregando el enlace del mapa en el código HTML de la web.

El mapa ofrece un botón que muestra la descripción del mapa, el número de visitas por parte de los usuarios y un listado de todos los marcadores en el mapa (ver Figura 10).

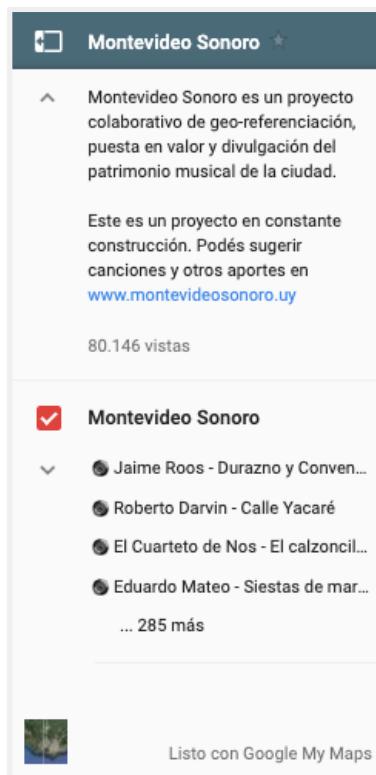


Figura 10. Información general Montevideo Sonoro

Este mapa cuenta con cerca de 300 marcadores distribuidos únicamente por la ciudad de Montevideo. El único evento de interacción que reconocen los marcadores son los de clic sobre ellos, cuando se realiza dicha acción, la información de la canción es desplegada en una ventana emergente que muestra: el título, el autor, el motivo del por qué esa canción está en ese lugar, un botón para ver la ruta hasta la canción y en algunas canciones se puede encontrar contenido multimedia para ver y reproducir (ver Figura 11).

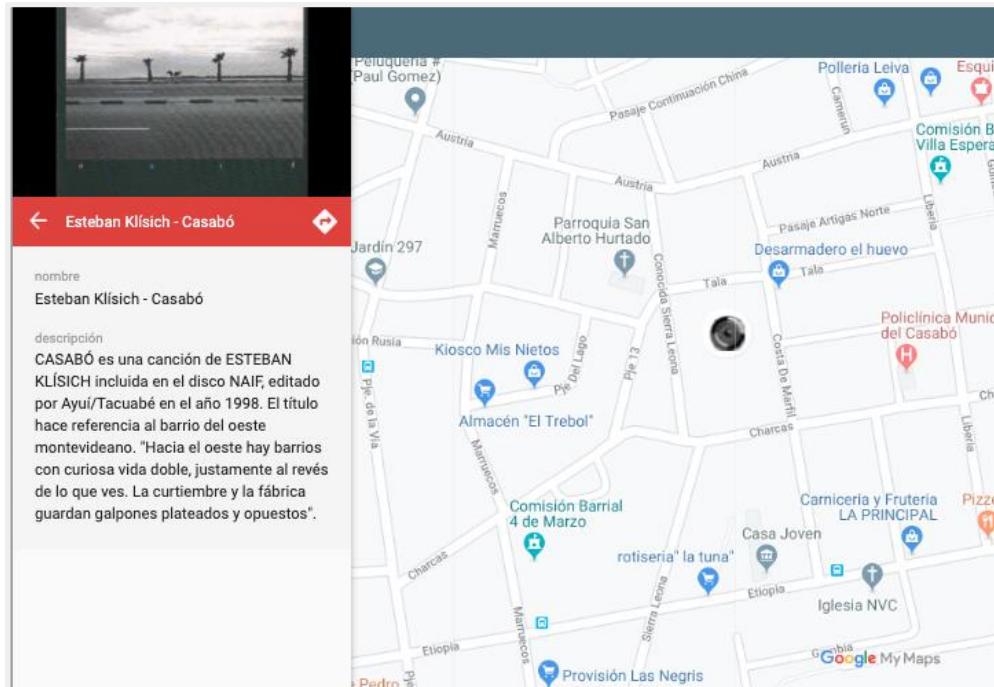


Figura 11. Información del marcador seleccionado en sitio Montevideo Sonoro

2.5.2. Puntos Sobre el Plano

Como se mencionó en la sección 2.4.2, la banda porteña geolocaliza su trabajo musical en un mapa web titulado “Puntos Sobre el Plano”.

Al ingresar se puede apreciar un mapa situado en Valparaíso, Chile, con marcadores amarillos que indican los lugares que fueron inspiración para las canciones de la banda, cada uno de los cuales conduce al audio y video de una canción diferente. Además, en la pantalla principal se pueden ver enlaces a las redes sociales e información de los colaboradores y creadores (ver Figura 2).

La plataforma web diseñada utiliza herramientas como Google Maps, WordPress y YouTube.

Actualmente los nueve marcadores poseen dos tipos de eventos de interacción, el primero muestra el nombre de la canción al pasar el cursor del *mouse* sobre el marcador (ver Figura 12), mientras que el segundo evento muestra la información de la canción al hacer clic sobre éste (ver Figura 13), la información desplegada en el último caso es: título de la canción, nombre del artista, video de la canción en YouTube y al final la descripción e historia de la canción.

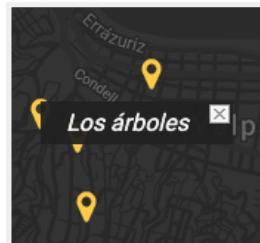


Figura 12. Marcadores en el mapa web de Puntos Sobre el Plano

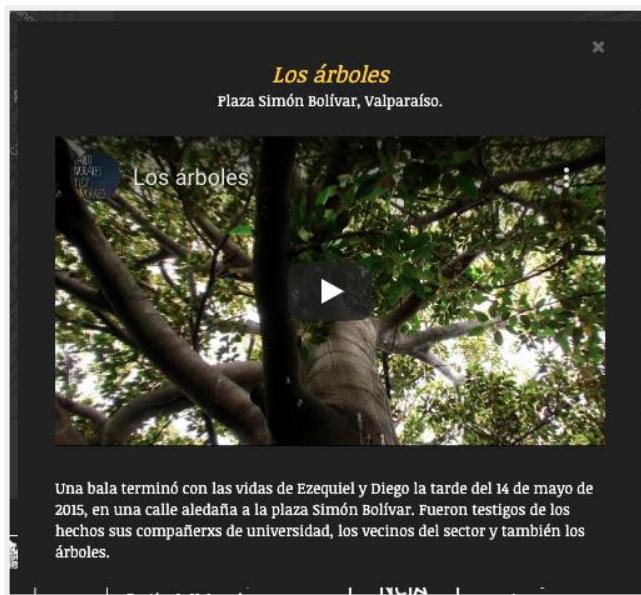


Figura 13. Información del marcador seleccionado en Puntos Sobre el Plano

3. DISEÑO

3.1. Requerimientos de diseño

Los requerimientos de diseño de *software* han sido recopilados a través de reuniones con el patrocinante del proyecto. Siendo los primeros requerimientos en ser identificados los que engloba en su conjunto la idea principal del proyecto, que es la georreferenciación musical, mientras que los demás requerimientos fueron identificados a través de las reuniones semanales durante la etapa de desarrollo.

Para representar los requerimientos se utilizaron historias de usuario, los cuales son una descripción de una funcionalidad que se debe incorporar en el sistema a desarrollar.

Primero, se identificaron todos los posibles usuarios que harían uso de la aplicación web, esto permitió redactar las historias de usuario en una hoja de cálculo de Google siguiendo el formato “Como [usuario] quiero [característica/funcionalidad] para poder [finalidad]”. Como siguiente paso se agruparon las historias de usuario que podían ser unificadas. Finalmente, se consideró que, siendo un desarrollo de un producto de *software*, en el que se aplicarían metodologías ágiles de desarrollo, sería mejor priorizar los requisitos para poder planificar iteraciones y entregables del producto de la mejor manera.

Para la tarea de priorizar los requerimientos se utilizó el método “MoSCoW”, acrónimo cuyas siglas en inglés significan *Must Have* (debe tener), *Should Have* (debería tener), *Could Have* (podría tener) o *Would not Have* (no tendrá por esta ocasión). En otras palabras, este método de priorización nos entrega un rápido análisis de las funcionalidades que permiten entregar mayor valor, lo que permite una buena planificación y un desarrollo más eficiente. Lo más interesante de este método con respecto a otros es que la escala utilizada entrega un fácil y rápido entendimiento de la criticidad de cada funcionalidad, ya que, mientras otros métodos nos proponen una escala numérica, MoSCoW nos aporta un valor semántico de lo que realmente es importante. La Tabla 2 muestra la descripción de cada una de las categorías del método MoSCoW.

Tabla 2. Descripción de categorías del método MoSCoW¹⁴

Etiqueta	Descripción
M - <i>MUST</i> - Debe tener	Se tratan de requisitos totalmente imprescindibles que tienen que estar incluidos ya que si no se llevan a cabo el proyecto no puede salir adelante
S - <i>SHOULD</i> - Debería Tener	Se tratan de requisitos que deberían de llevarse a cabo si es posible. Estos requisitos no se tratarían de <i>stoppers</i> para poner el proyecto en marcha si no se llegan a tener a tiempo. También conocidos como requisitos “Nice to Have”
C - <i>COULD</i> - Podría tener	Se tratan de requisitos que podrían incluirse si no afecta a nada más, es decir, son requisitos que sería bueno tener y podrían incluirse porque no cuesta demasiado implementarlas. Estos requisitos podrían quedarse en el <i>backlog</i> para que en una fase posterior se implementen
W - <i>WON'T</i> - No tendrá esta vez	Se tratan de requisitos que no se implementarán en la fase que se está planteando, pero lo pueden estar en un futuro. Estos requisitos son excluidos del alcance, pero no por ello no son importantes, simplemente que no aplican en la fase que se está planteando

3.1.1. Arquetipos de usuario

La identificación y definición de los usuarios permiten generar arquetipos, lo cual es fundamental porque ayuda a visualizar de manera rápida a los usuarios para quienes se está diseñando. Además, ayuda a orientar estrategias de una metodología centrada en el cliente, lo que puede asegurar un cierto grado de éxito en etapas posteriores del proyecto.

Los usuarios se definen en la Tabla 3, junto a las necesidades que presentan, además de indicar algunas de las soluciones actuales que las satisfacen.

Tabla 3. Necesidades de los usuarios

Usuario	Necesidad	Solución Actual
Ciudadanos o lugareños	Conocer la música que los rodea	Emisoras radiales de la zona, eventos locales y prensa escrita
Viajeros amantes de la música	Conocer las canciones que fueron inspiradas en los lugares que visita	<i>Blogs</i> , páginas web y eventos locales
Artistas musicales	Promover su música	Plataformas musicales, redes sociales y eventos
Fanáticos de la música	Compartir y aumentar su conocimiento musical	<i>Blogs</i> , redes sociales, páginas web, comunicación verbal y prensa escrita
Encargados del turismo	Fomentar la cultura musical de una localidad	Redes sociales, <i>blogs</i> , folletos, prensa escrita y emisoras radiales

¹⁴ <https://adrianalonso.es/project-management/priorizacion-requisitos-software-con-moscow/>

Los arquetipos de usuario son definidos en la Tabla 4.

Tabla 4. Arquetipos de usuario

Arquetipo	Objetivo al usar el sistema	Actividades dentro del sistema
Ciudadanos o lugareños	Conocer e interiorizarse sobre la cultura musical local	Navegar por la aplicación y el mapa para poder filtrar por localidad, para así aprender sobre la música que los rodea
Viajeros amantes de la música	Conocer las canciones que menciona la localidad o cercanías del lugar que está visitando o que visitará	Al utilizar la aplicación con su GPS activado, MusycGo le indicará las canciones cercanas a su posición
Artistas musicales	Dar a conocer su arte musical, expresando a través de sus letras las vivencias y pensamientos que quieran transmitir al mundo	Al utilizar la aplicación pueden realizar aportes y colaboraciones, de esta forma llegarán a más personas, mostrando su trabajo y lo que quieren expresar con su arte
Fanáticos de la música	Conocer más sobre los lugares que inspiraron canciones, además de entregar su conocimiento a los usuarios, a través de sus colaboraciones justificadas	Aprender más sobre la música local o de su agrado. Pueden ingresar como un usuario registrado para poder compartir su conocimiento musical a través georreferenciaciones, justificando su decisión.
Encargados del turismo	Fomentar la cultura musical de una localidad	Compartir la música de una localidad con las demás personas a través de redes sociales. Realizar contribuciones con música de artistas locales.

3.1.2. Perfil de comportamiento de usuario

Cuando un usuario ingresa al sistema MusycGo, puede ser: **usuario explorador**, **usuario colaborador** y **usuario moderador**, la adopción de alguno se limita al interés o conocimiento que presenta dicho usuario al ingresar, o simplemente porque tiene los privilegios necesarios para ser un usuario moderador.

Comportarse como un usuario explorador se refiere a que el usuario si bien pueda tener conocimiento de canciones que hablan de lugares, no desea contribuir por el momento, por lo que su interés es solo escuchar, ver y descubrir. Un usuario colaborador es el que quiere difundir su conocimiento a la comunidad, a través de georreferenciaciones, incluso puede editar la información de las canciones. Finalmente tenemos al usuario moderador, por sus privilegios dentro del sistema puede desvincular canciones que carezcan de sentido o justificaciones para considerarla perteneciente a algún lugar.

Hay que mencionar que la aplicación se comporta de la misma manera para ambos perfiles, por lo que adoptar uno u otro no implica tener más o menos funcionalidades a su disposición. Aunque existe una característica que tendrán en especial los usuarios colaboradores (si se encuentra registrado en el sistema), y es que pueden realizar aportes

registrando su nombre de usuario, luego pueden ver el reconocimiento a su aporte en la información de la canción registrada, además de ser reconocido en las estadísticas generales de la aplicación, donde pueden posicionarse en un *ranking* dado el número de colaboraciones realizadas. También para el usuario moderador la vista cambia levemente, ya que existen funcionalidades para poder eliminar el contenido considerado inadecuado.

3.1.3. Requerimientos funcionales

Para el presente proyecto de título se identificaron requerimientos que constituyen las funcionalidades o lo que el sistema debe hacer, para los módulos: Mapa Principal, Ingreso de Canciones, Buscador General, Comentarios, Estadísticas, Perfil Maestro de Canciones, Perfil de Usuario, Perfil de Lugar de Canción y Notificaciones.

Se presentan los requerimientos agrupados por módulos en las Tablas 5, 6, 7, 8, 9, 10, 11, 12 y 13 respectivamente, además se indica el estado de cumplimiento de cada uno.

Tabla 5. Requerimientos funcionales para el módulo: Mapa Principal

Como [usuario]	Quiero [característica]	Para poder [finalidad]	Prioridad (MoSCoW)	Estado
Viajero amante de la música	Un listado de las canciones cercanas a mi ubicación	Conocer la cultura musical que me rodean	Could	Cumplido
Fanático de la música	Acceder a plataformas externas de música (Spotify y YouTube)	Reproducir la canción en el marcador seleccionado	Must	Cumplido
Fanático de la música	Obtener información de la canción y del autor, a través del marcador seleccionado	Conocer la información básica de la canción	Must	Cumplido
Fanático de la música	Usar mi cuenta de Spotify	Reproducir las canciones completas	Should	Cumplido
Fanático de la música	Usar la herramienta en mi idioma (español, inglés o portugués)	Entender el significado del texto que se muestra en pantalla	Should	Cumplido
Fanático de la música	Filtrar los resultados de las canciones según los géneros musicales	Mostrar sólo los resultados de mi interés	Should	Cumplido
Fanático de la música	Filtrar los resultados de las canciones según el idioma de la letra	Mostrar sólo los resultados de un idioma en particular	Could	Cumplido
Fanático de la música	Filtrar los resultados de las canciones según el autor/compositor de la música	Mostrar sólo los resultados del autor/compositor en particular	Should	Cumplido
Fanático de la música	Compartir la música y ubicación de la canción que estoy escuchando (o que escuche)	Dar a conocer una canción interesante para mí	Must	Cumplido
Fanático de la música	Ver rápidamente el fragmento de la letra o justificativa correspondiente a la georreferenciación desde el marcador seleccionado	Conocer por qué la canción se encuentra en ese lugar	Should	Cumplido
Viajero amante de la música	Recalcular mi posición GPS en el mapa	Ver las nuevas canciones según mi posición actual	Should	Cumplido
Viajero amante de la música	Ver las canciones que hacen referencia a un lugar cercano a mi ubicación	Conocer mi entorno musical	Must	Cumplido

Tabla 5. (Continuación)

Como [usuario]	Quiero [característica]	Para poder [finalidad]	Prioridad (MoSCoW)	Estado
Fanático de la música	Obtener una lista de canciones georreferenciadas en determinado país	Conocer y reproducir las canciones que hacen referencia a un país en particular	<i>Should</i>	Cumplido
Explorador fanático de la música	Recomendaciones según donde estoy	Aprender más sobre el lugar	<i>Must</i>	Cumplido
Fanático de la música	Obtener una lista de canciones georreferenciadas a partir de mi ubicación	Conocer y reproducir las canciones que hacen referencia a un lugar cercano a mi ubicación	<i>Should</i>	Cumplido
Fanático de la música	Reproducir la canción dentro de la aplicación	Acceder rápidamente a la canción, y así entretenarme y escuchar el fragmento de la letra que hace referencia al lugar	<i>Should</i>	Cumplido
Fanático de la música	Saber el autor de la georreferenciación de determinada canción en determinada ubicación	Conocer quien realizó el aporte	<i>Should</i>	Cumplido
Fanático de la música	Visualizar el planisferio con sus <i>clusters</i> de canciones por país	Conocer todas las canciones georreferenciadas en cada país	<i>Should</i>	Cumplido
Explorador fanático de la música	Ver las unidades político-administrativas	Tener una mejor noción de donde “pertenece” cada canción	<i>Could</i>	Cumplido
Explorador fanático de la música	Ver información sobre la simbología del mapa	Tener un mayor entendimiento de los marcadores y botones presentes en el mapa	<i>Could</i>	Cumplido

Tabla 6. Requerimientos funcionales para el módulo: Ingreso de Canciones

Como [usuario]	Quiero [característica]	Para poder [finalidad]	Prioridad (MoSCoW)	Estado
Ciudadano o Lugareño	Georreferenciar las canciones de mi región	Mostrar las canciones de mi región a las demás personas	<i>Could</i>	Cumplido
Artista musical	Georreferenciar mi música	Dar a conocer mi trabajo	<i>Could</i>	Cumplido
Fanático de la música	Georreferenciar varios lugares en un mismo aporte	Evitar realizar un aporte para cada lugar en donde se encuentra la canción	<i>Must</i>	Cumplido
Fanático de la música	Georreferenciar músicas	Dar a conocer mi erudición	<i>Could</i>	Cumplido
Fanático de la música	Georreferenciar más lugares a canciones que ya están registradas	Completar aún más la pertenencia de la canción en el mundo	<i>Should</i>	Cumplido
Fanático de la música	Destacar (escribir) la parte de la letra referente a la ubicación seleccionada	Justificar la pertenencia de la canción en ese lugar	<i>Should</i>	Cumplido
Fanático de la música	Subir la letra de la canción en mi aporte	Completar la información de la canción que estoy ingresando	<i>Should</i>	Cumplido
Colaborador fanático de la música	Seleccionar alguna opción u observación, sobre la razón de geolocalización	Explicar rápidamente la razón de mi aporte	<i>Could</i>	Cumplido

Tabla 7. Requerimientos funcionales para el módulo: Buscador General

Como [usuario]	Quiero [característica]	Para poder [finalidad]	Prioridad (MoSCoW)	Estado
Fanático de la música	Una lista de todas las canciones en la aplicación	Conocer la música registrada	<i>Should</i>	Cumplido
Fanático de la música	Ingresar el nombre de una canción en un buscador	Saber si existen coincidencias y de encontrarse algún resultado poder verlo en el mapa	<i>Must</i>	Cumplido
Fanático de la música	Ingresar el nombre de un lugar	Ver las canciones que hacen referencia a ese lugar	<i>Must</i>	Cumplido

Tabla 8. Requerimientos funcionales para el módulo: Comentarios

Como [usuario]	Quiero [característica]	Para poder [finalidad]	Prioridad (MoSCoW)	Estado
Fanático de la música	Calificar la calidad de la canción per se	Opinar sobre aspectos de la justificativa o georreferenciación	Should	Cumplido

Tabla 9. Requerimientos funcionales para el módulo: Estadísticas

Como [usuario]	Quiero [característica]	Para poder [finalidad]	Prioridad (MoSCoW)	Estado
Fanático de la música	Aparecer en un listado tipo <i>ranking</i> con mi número de colaboraciones	Ser reconocido por mis contribuciones	Should	Cumplido

Tabla 10. Requerimientos funcionales para el módulo: Perfil Maestro de Canciones

Como [usuario]	Quiero [característica]	Para poder [finalidad]	Prioridad (MoSCoW)	Estado
Fanático de la música	Saber datos geográficos de determinada canción	Conocer a qué lugares está relacionada	Could	Cumplido
Fanático de la música	Editar atributos de la canción	Modificar datos incorrectos o para mejorar los que ya se encuentran	Should	Cumplido

Tabla 11. Requerimientos funcionales para el módulo: Perfil de Usuario

Como [usuario]	Quiero [característica]	Para poder [finalidad]	Prioridad (MoSCoW)	Estado
Fanático de la música	Registrarme con alguna de mis cuentas de Gmail, Facebook o Twitter	Facilitar el ingreso de mis datos en la aplicación	Should	Cumplido
Fanático de la música	Tener una cuenta registrada	Guardar en favoritos los lugares con mis canciones preferidas	Should	Cumplido
Fanático de la música	Exhibir mi perfil a los demás usuarios	Dar a conocer mis contribuciones a los demás	Should	Cumplido

Tabla 12. Requerimientos funcionales para el módulo: Perfil de Lugar de Canción

Como [usuario]	Quiero [característica]	Para poder [finalidad]	Prioridad (MoSCoW)	Estado
Fanático de la música	Ver el fragmento de la letra o justificativa correspondiente a la georreferenciación	Conocer por qué la canción se encuentra en ese lugar	Should	Cumplido
Fanático de la música	Compartir la música y ubicación de la canción que estoy escuchando (o que escuche)	Dar a conocer una canción interesante para mí	Must	Cumplido
Fanático de la música	Editar la justificativa de georreferenciación de determinada canción-lugar	Corregir y/o especificar la justificativa de la canción-lugar	Should	Cumplido
Fanático de la música	Visualizar la cantidad de “likes” o “favoritos” de cada canción	Saber la popularidad de cada canción	Could	Cumplido
Fanático de la música	Calificar la pertenencia de determinada canción al lugar	Votar según mi apreciación sobre el nivel de pertenencia de la canción al lugar	Should	Cumplido
Fanático de la música	Editar atributos del lugar y/o de la canción	Modificar datos incorrectos o para mejorar los que ya se encuentran	Should	Cumplido

Tabla 13. Requerimientos funcionales para el módulo: Notificaciones

Como [usuario]	Quiero [característica]	Para poder [finalidad]	Prioridad (MoSCoW)	Estado
Fanático de la música	Recibir notificaciones de cambios en las canciones ingresadas por mí	Conocer mis georreferenciaciones que han sido modificadas	Could	Cumplido

3.1.4. Requerimientos no funcionales

Los requisitos no funcionales especifican las propiedades emergentes que el sistema debe tener, características de: diseño, rendimiento, restricciones de tiempo, estándares de seguridad, entre otros, se presentan en las Tablas 14, 15, 16 y 17, además se muestra el estado de cumplimiento para cada requerimiento.

Tabla 14. Requerimientos no funcionales de seguridad

Requisito	Descripción	Prioridad	Estado
Protocolo SSL para la navegación	Se ofrece una capa que proporciona comunicaciones seguras entre el navegador web y MusycGo	<i>Must</i>	Cumplido
Respaldo diario de la base de datos	Realizar día a día una copia de los datos de usuarios y de la música de MusycGo	<i>Should</i>	Cumplido
Expiración de sesión de usuario	Luego de un tiempo determinado la plataforma expira la sesión automáticamente	<i>Could</i>	Cumplido
Conexión segura a la API	El sitio web soporta una conexión segura a la API a través de un protocolo SSL	<i>Should</i>	Cumplido
Contraseñas no disponibles por vía API	Asegurar que no se pueda acceder a las contraseñas a través de la API de MusycGo	<i>Must</i>	Cumplido

Tabla 15. Requerimientos no funcionales de usabilidad

Requisito	Descripción	Prioridad	Estado
Diseño <i>responsive</i>	Adaptación automática de los elementos de la aplicación según el dispositivo en donde se esté accediendo (Anexo C muestra el diseño en <i>smartphone</i>)	<i>Should</i>	Cumplido
Multilenguaje	La aplicación es legible y entendible en tres idiomas (español, portugués e inglés), dada la selección del usuario	<i>Should</i>	Cumplido
Conservar el diseño	Utilizar colores y estilos gráficos concordantes en toda la aplicación	<i>Should</i>	Cumplido
El sistema transparenta la política de uso	Existe un apartado o enlace dentro de la plataforma que permita leer las políticas de uso	<i>Could</i>	No cumplido
El navegador no requiere <i>plugins</i> o instalaciones previas	Para el uso de MusycGo no se requiere ninguna instalación en el navegador desde el que se accede	<i>Should</i>	Cumplido

Tabla 16. Requerimientos no funcionales de fiabilidad

Requisito	Descripción	Prioridad	Estado
Madurez componentes del sistema	Herramientas utilizadas para el diseño, construcción, pruebas e implementación reconocidas que tengan más de tres años en el mercado	<i>Could</i>	Cumplido
Disponibilidad	El 99,5% del tiempo MusycGo se encuentra en estado de actividad en un año determinado	<i>Must</i>	Cumplido

Tabla 17. Requerimientos no funcionales de rendimiento

Requisito	Descripción	Prioridad	Estado
Tiempo de carga del mapa	10 segundos	<i>Must</i>	Cumplido
Tiempo de búsqueda de canciones según una cadena de caracteres	2 segundos	<i>Must</i>	Cumplido
Tiempo de guardado de un nuevo aporte	3 segundos	<i>Must</i>	Cumplido
Tiempo de modificación de un aporte	3 segundos	<i>Must</i>	Cumplido
Tiempo de eliminación de un aporte	3 segundos	<i>Should</i>	Cumplido
Tiempo de carga de las estadísticas	4 segundos	<i>Must</i>	Cumplido
Tiempo de respuesta de la autenticación	5 segundos	<i>Should</i>	Cumplido
Funcionalidad en distintos navegadores web	La aplicación puede ser accedida y utilizada en navegadores Safari, Google Chrome, Firefox, entre otros. (Probado en Google Chrome v79 en Windows, Google Chrome v80 en Mac, Firefox v73.0.1 en Mac y Safari v13.0.4 en Mac)	<i>Should</i>	Cumplido

3.2. Diseño de interfaz

El proyecto MusycGo tuvo sus inicios bajo el nombre de “GucApp” a mediados del año 2018, en ese entonces se propuso realizar una prueba de concepto para poder validar su factibilidad. Para esta versión preliminar se incorporaron las funcionalidades que transmitían la idea de posicionar música en los lugares que las inspiraron.

El resultado de la prueba de concepto se muestra en la Figura 14, la cual muestra básicamente un mapa con los marcadores de las canciones que han sido preconfigurados en el sistema. En la misma figura se pueden ver dos filtros en el costado izquierdo, que permiten seleccionar el radio de alcance y el lugar (registrado previamente de forma manual en el código fuente), luego se pueden ver filtradas las canciones cercanas al Mercado Fluvial de Valdivia en un radio de 800 metros. También tiene la posibilidad de seleccionar un marcador en el mapa para obtener información extendida de la canción seleccionada. Además, en la Figura 15 se muestra un apartado de recomendaciones de canciones cercanas a la ubicación seleccionada.

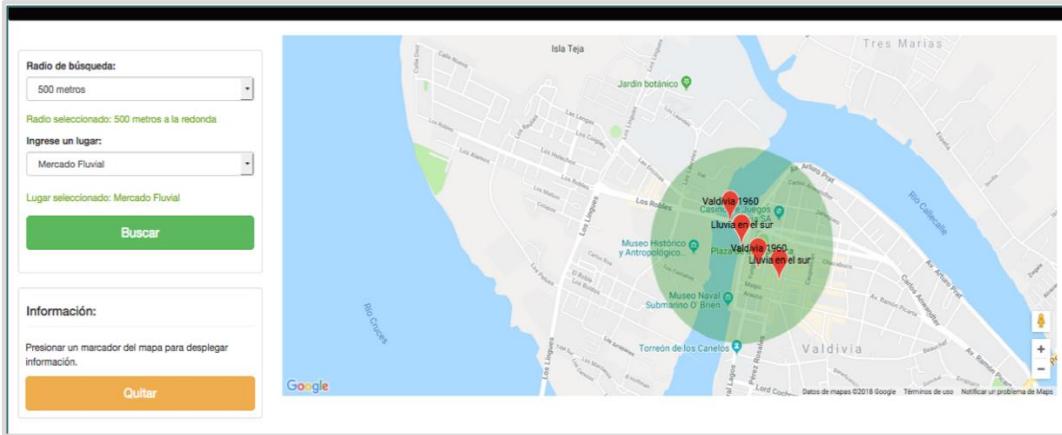


Figura 14. Vista principal prueba de concepto



Figura 15. Recomendaciones canciones cercanas en prueba de concepto

En base a los arquetipos de usuarios mencionados en Tabla 4 se pudo crear una propuesta de diseño, donde se esquematizan las acciones que ellos pueden hacer, además de poder crear una arquitectura de información, forma de navegación y vistas.

En la versión actual de este prototipo se puede distinguir un rediseño sustancial de lo que fue la prueba de concepto, se modificó la forma en que se muestran los elementos en la pantalla (filtros, mapa, encabezado, etc.), dejando que el mapa ocupe el ancho completo de la ventana. Por otra parte, se agregaron nuevas ventanas, se cambió el diseño del mapa y de sus marcadores, entre otros elementos que siguen un diseño y una paleta de colores.

3.3. Diseño del software

El desarrollo de una aplicación web involucra entre muchas cosas el uso de diferentes herramientas y tecnologías, gestión de una base de datos, operaciones del lado del servidor (*backend*), así como el lado del cliente (*frontend*) y finalmente, el procesamiento, despliegue y visualización de los datos que provienen del servidor. Por eso el uso de un *stack* ayuda a organizar todas las herramientas, *frameworks*, tecnologías y la estructura del proyecto en general. Para el presente proyecto se utilizó el MEAN Stack (MongoDB, ExpressJS, Angular y NodeJS), el cual se describe en la siguiente sección (ver Figura 16).

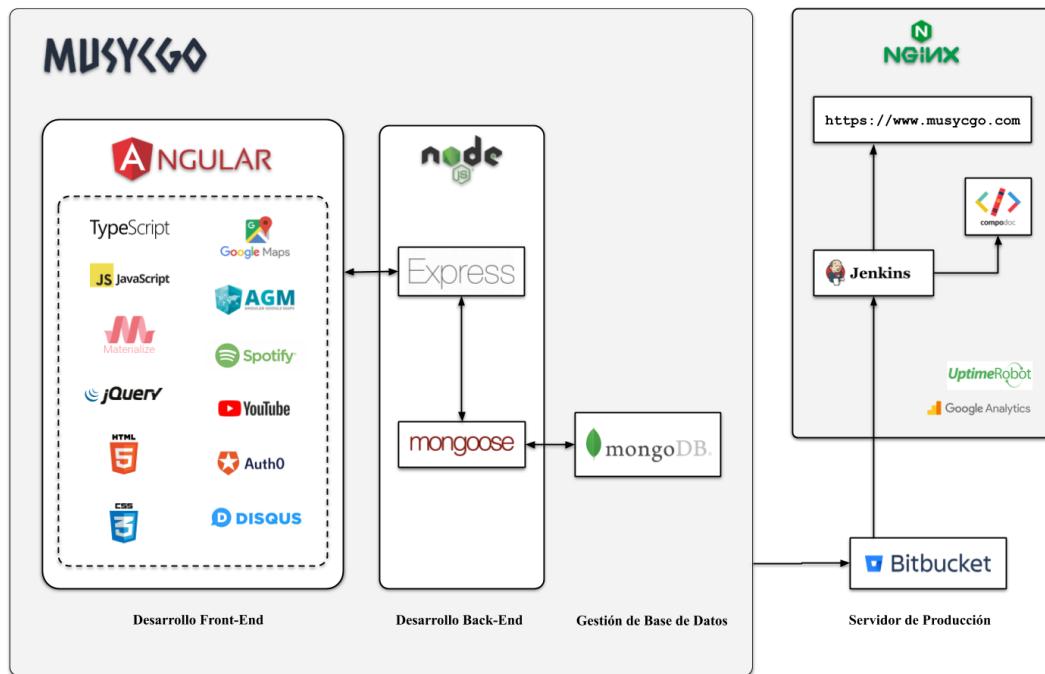


Figura 16. Diagrama de tecnologías de MusycGo

3.3.1. MEAN Stack

MEAN está integrada por las tecnologías: MongoDB, Express, Angular y NodeJS, en su conjunto forman una plataforma llamada “*Stack*” y estableció una sólida base para llevar a cabo el desarrollo de MusycGo.

Stack MEAN soporta el lenguaje JavaScript tanto en el lado del servidor, como en el del cliente, eso soluciona un problema conocido como “fatiga del lenguaje”, es decir, al trabajar con el mismo lenguaje constantemente para el desarrollo de las diversas partes del proyecto, permite pasar de una plataforma a otra sin tener que cambiar de léxico o sintaxis, a la vez que permite reutilizar el conocimiento adquirido, evitando fatiga mental.

Optar por el lenguaje JavaScript se fundamenta por varios aspectos, se listan las siguientes ventajas relevantes para el presente proyecto:

- Es soportado por múltiples navegadores.
- Versatilidad, ya que es útil para desarrollar páginas dinámicas y aplicaciones web, permite hacerlas interactivas.
- Es un lenguaje que permite trabajar en modo *Full Stack* (*backend* y *frontend*) en cualquier tipo de desarrollo.
- Personalizar mapas digitales a través de la API de Google Maps con una documentación basada en JavaScript.
- Código de programación legible y explicativo.

- Una gran comunidad y proyectos que lo validan.

Además, utilizar este lenguaje se debe a su alto éxito y popularidad presentados en los últimos años. Según la encuesta anual (Stack Overflow, 2019), en la categoría de tecnologías, JavaScript fue por séptimo año consecutivo el lenguaje de programación más popular y utilizado tanto por profesionales (69,7%) y no profesionales (67,8%) en el año 2019.

Tomando la Figura 17 como referencia, se puede apreciar el concepto y la interacción entre las tecnologías del MEAN Stack, cuya breve descripción de cada paso es la siguiente:

1. Cuando un usuario realiza una solicitud a través del navegador, Angular la procesa en primera instancia.
2. La solicitud es analizada por el servidor, donde se encuentra NodeJS.
3. Luego ExpressJS realiza la solicitud a la base de datos.
4. MongoDB (base de datos) recupera los datos y los envía como respuesta a ExpressJS.
5. ExpressJS devuelve la respuesta a NodeJS.
6. NodeJS devuelve la respuesta a Angular y luego es mostrado al usuario en pantalla.

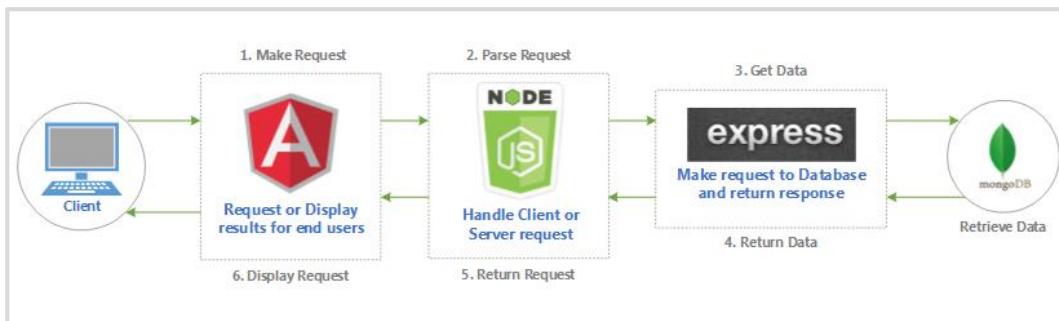


Figura 17. Concepto MEAN Stack¹⁵

3.3.2. MongoDB

MongoDB¹⁶ es una base de datos de tipo NoSQL orientada a documentos y de esquema libre que proporciona alta capacidad de almacenamiento, velocidad en las consultas, permite trabajar con un gran volumen de datos con un alto rendimiento, además de ser escalable.

Al estar orientada a documentos, MongoDB se contrapone al modelo relacional, ya que no existen tablas, sino documentos en formato JSON¹⁷ en el caso de este proyecto.

¹⁵ <https://dzone.com/articles/6-things-you-want-to-know-about-mean-stack>

¹⁶ <https://www.mongodb.com/>

¹⁷ JSON de siglas en inglés *JavaScript Object Notation* (Notación de Objeto de JavaScript).

Las bases de datos NoSQL son aquellas que no se ajustan al modelo de base de datos relacionales. Este tipo de base de datos no usan SQL como lenguaje de consultas, no requieren estructuras fijas como tablas, hacen uso amplio de la memoria principal de máquina donde se encuentre alojada y son ideales para resolver el problema de los altos volúmenes de información, consultas y transacciones diarias. Por esta última característica las bases de datos NoSQL son recomendables para el ámbito de las aplicaciones web con muchos usuarios.

Para proyectos que consideren la escalabilidad de su plataforma, las bases de datos orientadas a documentos pueden conservar la simplicidad que en el caso de las bases de datos relacionales no se puede tener debido a la compleja interacción que puede ocurrir entre sus esquemas cuando el sistema comienza a requerir algún tipo de escalamiento.

De las bases de datos NoSQL orientada a documentos, MongoDB es la más sofisticada y utilizada hoy en día, así evidencia el “Ranking DB-Engines” (DB-Engines, 2020).

En la Figura 18 se muestra la gran popularidad que ha mantenido MongoDB durante enero y febrero del año 2020, en donde la puntuación del primer lugar es casi siete veces mayor al segundo puesto de la lista. Además, en la Figura 19 podemos ver como MongoDB ha liderado con amplia ventaja respecto a los demás.

Rank			DBMS	Database Model	Score		
Feb 2020	Jan 2020	Feb 2019			Feb 2020	Jan 2020	Feb 2019
1.	1.	1.	MongoDB	Document, Multi-model	433.33	+6.37	+38.24
2.	2.	2.	Amazon DynamoDB	Multi-model	62.14	+0.12	+7.19
3.	3.	3.	Couchbase	Document, Multi-model	32.16	+0.12	-3.42
4.	4.	4.	Microsoft Azure Cosmos DB	Multi-model	31.95	+0.44	+7.09
5.	5.	5.	CouchDB	Document	18.13	-0.24	-1.87
6.	↑ 7.	↑ 7.	Firebase Realtime Database	Document	12.35	+0.07	+1.92
7.	↓ 6.	↓ 6.	MarkLogic	Multi-model	12.25	-0.11	-2.74
8.	8.	8.	Realm	Document	8.85	+0.62	+1.97
9.	9.	↑ 15.	Google Cloud Firestore	Document	6.21	+0.55	+2.45
10.	↑ 11.	↓ 9.	OrientDB	Multi-model	4.94	-0.17	-1.11

Figura 18. Ranking DBMS¹⁸

¹⁸ <https://db-engines.com/en/ranking/document+store>

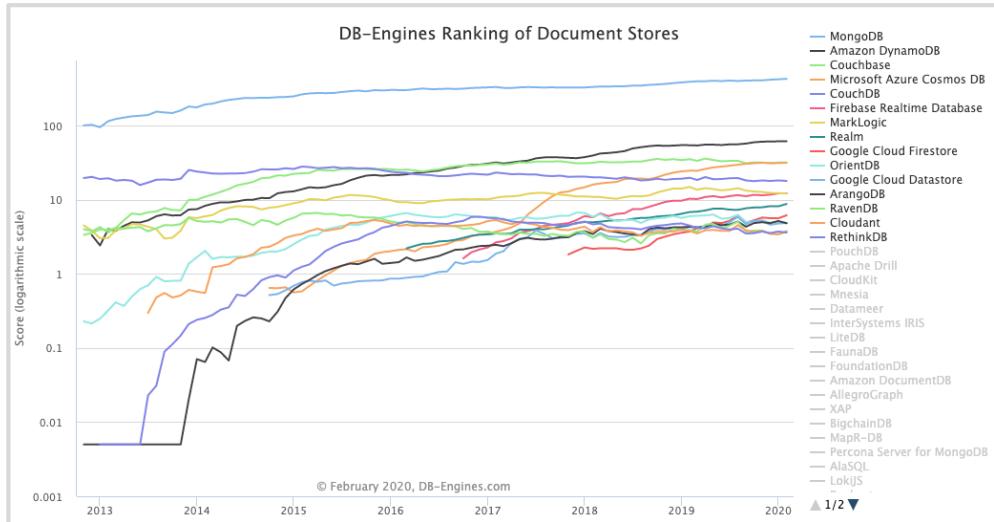


Figura 19. DB-Engines ranking de base de datos orientadas a documentos¹⁹

MongoDB es la base de datos que utiliza MusycGo para almacenar los datos correspondientes a usuarios y las canciones.

En la Figura 20 se puede ver la información de la base de datos de MusycGo (fecha 24 de marzo del 2020), la cual contiene dos colecciones de datos: “musics” y “users”, ambos están conformados por documentos o registros que contienen todos los datos de la música y de los usuarios registrados en la plataforma, además se indica el tamaño total de las colecciones.

Collection Name	Documents	Avg. Document Size	Total Document Size	Num. Indexes	Total Index Size	Properties
musics	164	2.2 KB	357.0 KB	1	24.6 KB	
users	7	1.5 KB	10.3 KB	1	24.6 KB	

Figura 20. Información de la base de datos de MusycGo

3.3.3. Express

Express²⁰ es un *framework* para NodeJS, es decir, es una plataforma para construir aplicaciones web en el lado del servidor usando NodeJS.

Express ayuda a organizar las aplicaciones web, ya que permite la asignación de rutas en función de los métodos HTTP²¹ y las URL²², el manejo de solicitudes y vistas, necesarios para la creación de una API Rest. Cuando hablamos de una arquitectura Modelo-Vista-

¹⁹ https://db-engines.com/en/ranking_trend/document+store

²⁰ <https://expressjs.com/>

²¹ HTTP de siglas en inglés *Hypertext Transfer Protocol* (Protocolo de Transferencia de Hipertexto).

²² URL de siglas en inglés *Uniform Resource Locator* (Localizador de Recursos Uniforme).

Controlador, Express forma parte del controlador, donde se gestionan y procesan los datos y son enviados a las demás capas de la aplicación para permitir al usuario tener acceso a los datos e información.

Para MusycGo, Express tiene el trabajo de manejar las rutas de las solicitudes que realiza el cliente al servidor. En la Figura 21 se muestran las rutas para las peticiones a la base de datos de música.

```
var express = require('express');
var router = express.Router();

const music = require('../controllers/music.controller');

router.get('/', music.getMusics);
router.get('/looks', music.getLookMusics);
router.get('/hears', music.getHearMusics);
router.post('/get/by/iduser', music.getByIdUser);
router.get('/group/place', music.groupPlaces);
router.get('/group/country', music.groupCountry);
router.get('/group/musicuser', music.groupSongUsers);
router.get('/group/numfav', music.groupNumFav);
router.get('/group/musicfav', music.groupMusicFav);
router.get('/group/artist', music.groupArtists);
router.post('/get/by/nameartist/noeqcoord', music.getByNameArtistDistCoord);
router.post('/get/by/nameartist/eqcoord', music.getByNameArtCoord);
router.post('/get/by/nameartist/coord', music.getByNamArtCoordConcatMap);
router.post('/get/by/all/nameartist', music.getAllByNameArtist);
router.post('/get/by/linkspotify', music.getByLinkSpotify);
router.post('/get/by/one/linkspotify', music.getOneByLinkSpotify);
router.get('/get/by/id:id', music.getMusicById);
router.get('/get/by/country:country', music.getMusicByCountry);
router.post('/', music.createMusic);
router.get('/:id', music.getMusic);
router.put('/:id', music.editMusic);

module.exports = router;
```

Figura 21. Rutas para peticiones a la base de datos de música vía Express

3.3.4. Angular

Angular²³ es un *framework* de JavaScript, gratuito y de código libre, creado por Google para desarrollar el *frontend* de una aplicación web, es decir, la parte de la aplicación con la que los usuarios interactúan directamente.

Angular es una reinvención de la primera versión de AngularJS. Esta versión inicial revolucionó el desarrollo *frontend* y el desarrollo web avanzado. Luego de AngularJS, en septiembre del año 2016, Google lanzó la versión definitiva, Angular 2 (en estos días se conoce simplemente como Angular), en esta versión se hizo un cambio total del código, lo que conllevó a una arquitectura orientada totalmente a componentes.

Citando (Stack Overflow, 2018), Angular ocupa la segunda posición por ser la tecnología más utilizada el año 2018, por detrás de NodeJS.

²³ <https://angular.io/>

Angular es considerado el *framework* de desarrollo del lado del cliente (*frontend*) de MusycGo, en este se programan todas las funcionalidades, interacciones, estilos, manejo de mapas, despliegue de información, tablas, reproductores embebidos, etc.

La Figura 22 muestra un ejemplo del fragmento de código necesario para poder incorporar un mapa de Google en una aplicación de Angular, hay que señalar que en el presente proyecto existen más parámetros, variables y funciones, para que el mapa se muestre personalizado.

```
import { BrowserModule } from '@angular/platform-browser';
import { NgModule, Component } from '@angular/core';

import { AgmCoreModule } from '@agm/core';

@Component({
  selector: 'app-root',
  styles: [
    agm-map {
      height: 300px;
    }
  ],
  template: `
<agm-map [latitude]="lat" [longitude]="lng"></agm-map>
`
})
export class AppComponent {
  lat: number = 51.678418;
  lng: number = 7.809007;
}

@NgModule({
  imports: [
    BrowserModule,
    AgmCoreModule.forRoot({
      apiKey: 'YOUR_GOOGLE_MAPS_API_KEY'
    })
  ],
  declarations: [ AppComponent ],
  bootstrap: [ AppComponent ]
})
export class AppModule {}
```

Figura 22. Ejemplo de codificación en Angular²⁴

Angular consta de varios artefactos de desarrollo, a continuación, se detallan: Angular CLI²⁵ (intérprete de línea de comandos), TypeScript (lenguaje de programación), Componente, Directiva, Pipe y Servicio.

1. **Angular CLI:** Es un intérprete de línea de comandos de Angular que facilita el inicio de proyectos, ya que ofrece herramientas ya configuradas para tareas de depuración, *testing* y *deploy*. En un proyecto de desarrollo de *software* se necesitan realizar diversas tareas, por ejemplo: compilar el código fuente, gestionar dependencias, sistemas de empaquetado y compactado de código, analizadores de sintaxis, la creación de componentes, servicios y pipes en el caso de Angular, etc. Cada una de estas tareas pueden ser realizadas por Angular CLI.

²⁴ <https://angular-maps.com/>

²⁵ <https://cli.angular.io/>

2. **TypeScript**: Lenguaje de programación principal de Angular. Es un subconjunto de los lenguajes JavaScript/ECMAScript, que facilita el desarrollo por lo mencionado anteriormente sobre MEAN y el uso de JavaScript en el lado del cliente y en el servidor (sección 3.3.1). Si bien, TypeScript es el lenguaje principal de Angular, perfectamente se puede desarrollar el código a través del lenguaje JavaScript, e igualmente será interpretado por Angular.
3. **Componente**: Es una porción de código que se puede reutilizar dentro del proyecto actual o incluso en uno nuevo. Se considera como el artefacto protagonista en Angular, debido a que la aplicación se desarrolla en base a un árbol de componentes. Angular hace uso de los estándares de los componentes web, en otras palabras, son un conjunto de APIs que permiten crear nuevas etiquetas HTML personalizadas, reutilizables y auto contenidas, que incluso se puedan implementar en otros proyectos. Cada módulo que conforma la aplicación web hecho en Angular se forma en base a componentes independientes, los cuales se pueden insertar dentro de otros y asignar estilos propios para cada uno.
4. **Directiva**: Son funciones invocadas cuando el DOM²⁶ (*Document Object Model*) es compilado por Angular. Representan atributos en una etiqueta HTML. Estos atributos ofrecen variados comportamientos para los bloques del DOM. Las directivas pueden ser definidas por el desarrollador y/o utilizar las que nos entrega el *core* de Angular. El comportamiento dinámico entregado por una directiva puede ser muy variado, por ejemplo:
 - a. **ngIf**: Este condicional se utiliza cuando se desea mostrar u ocultar un elemento, la condición está determinada por el resultado de la expresión definida en la directiva, si la condición se cumple el elemento es mostrado, caso contrario el elemento se elimina del DOM.
 - b. **ngFor**: Esta directiva estructural repite una plantilla HTML (elemento o etiqueta a la que se adjunta la directiva) por cada elemento de una matriz (dato de tipo *array*).
 - c. **ngStyle**: Permite establecer propiedad de estilo a los elementos del DOM.
 - d. **ngClass**: Permite establecer clases CSS dinámicamente a los elementos del DOM.
5. **Pipe**: Son funciones (pequeñas) separadas de un componente, que se pueden utilizar en cualquier parte de la aplicación, es decir, para no escribir o programar la misma función para distintos componentes, se crea lo que se conoce como “pipe”, con esto se pueden realizar tareas que modifiquen la visualización de elementos. Por ejemplo, en MusycGo existen pipes para: modificar el formato de la fecha que viene desde el servidor para mostrarlos en pantalla, otro pipe aplica una transformación de metros a kilómetros, otro que obtiene la imagen de la bandera de un país y un pipe que permite cambiar el idioma de la aplicación.

²⁶ <https://medium.com/@janpierrsanchez/qu%C3%A9-es-el-dom-b0e041906b0a>

6. **Servicio:** Un servicio es un proveedor de datos. Estos se pueden conectar con el servidor para hacer peticiones HTTP a una API. Mantienen la lógica de acceso y operatividad de los datos. Los servicios son consumidos por los componentes y son los responsables de capturar los datos, guardarlos y luego realizar las operaciones correspondientes.

3.3.5. NodeJS

NodeJS²⁷, creado por Ryan Dahl en el año 2009, es una librería y un entorno de tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor, basado en el lenguaje de programación ECMAScript, asíncrono, con entrada y salida de datos en una arquitectura orientada a eventos (*Event Loop*) y basado en el motor JavaScript V8 de Google Chrome.

Este entorno de tiempos de ejecución incluye todo lo necesario para ejecutar programas escritos en JavaScript.

A comparación de otras técnicas de servicios web donde cada solicitud de conexión genera un subproceso mediante hilos independientes (ocupando memoria y afectando el rendimiento), NodeJS opera en un solo subproceso, llevando a cabo un modelo de entrada(s) sin bloqueo en la salida, lo que permite soportar muchísimas conexiones al mismo tiempo en un bucle que procesa un evento tras otro.

NodeJS fue formulado para generar sistemas escalables y permitir un elevado número de conexiones simultáneas con el servidor.

Según (Stack Overflow, 2019), en la categoría “Otros marcos, bibliotecas y herramientas”, NodeJS es el más utilizado según todos los encuestados (49,9%) y también a nivel de desarrolladores profesionales (50,4%) en el año 2019.

En el servidor web dentro del *hosting* reside un *script* de la instancia del servidor NodeJS, responsable del *backend* de MusycGo y con el cual se solicitan datos. Con la ayuda de la herramienta PM2²⁸ se inicia, gestiona y monitorea dicha instancia en particular. En la Figura 23 se puede observar la instancia de NodeJS de MusycGo y las peticiones que se realizan al visitar una página de la aplicación web.

²⁷ <https://nodejs.org/>

²⁸ <https://pm2.io/>

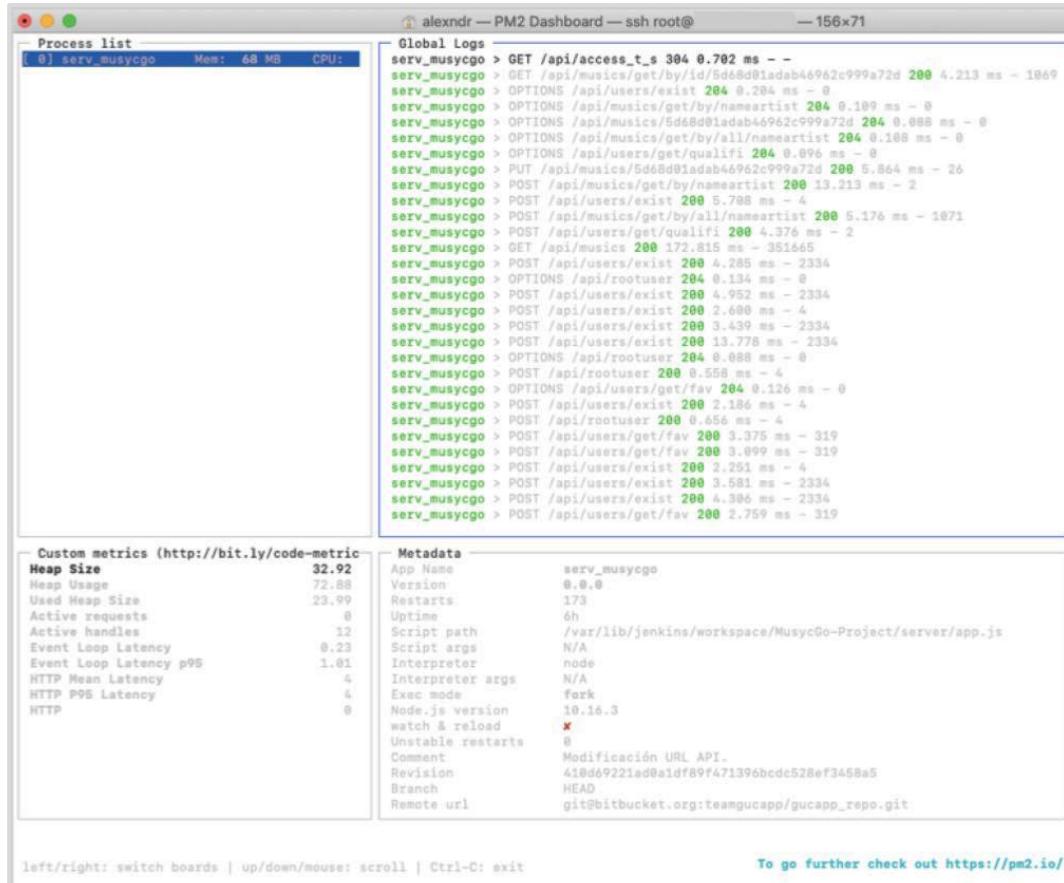


Figura 23. Monitoreo de la instancia de NodeJS vía PM2

3.3.6. NPM

NPM de sus siglas en inglés *Node Package Manager*, es un gestor de paquetes que da acceso a un conjunto de librerías gratuitas, generadas por la colaboración de los usuarios de su comunidad.

Este gestor ofrece la posibilidad de añadir funciones que no se encuentran por defecto en un proyecto, por ejemplo, en proyectos de Angular y NodeJS.

3.4. Integración Continua

La Integración Continua (CI, *Continuous Integration*) es una práctica del desarrollo de software donde se integra el código frecuentemente. Esta práctica es muy común en el ámbito del desarrollo ágil. El objetivo de este enfoque es trabajar en pequeños incrementos con el fin de lograr un proceso de desarrollo más efectivo y poder reaccionar con flexibilidad ante cualquier cambio. Los desarrolladores integran día a día su código (idealmente) en el repositorio donde está alojado el código fuente del proyecto, así todo el equipo puede tener conocimiento del progreso. Debido a los pequeños incrementos en

el código, la integración se realiza rápidamente. Si se descubren posibles errores en el proceso, se pueden localizar y solucionar en menos tiempo.

3.4.1. Prácticas esenciales de Integración Continua

Cuando se está utilizando el enfoque de la Integración Continua, se debe considerar una serie de prácticas, las cuales están plasmadas en los 11 principios que Martin Fowler describió para llevar a cabo con éxito una Integración Continua (Fowler, 2006).

Con la ayuda de las 11 prácticas de Integración Continua que postula Fowler, se puede realizar un diagnóstico para estar al tanto del estado de un proyecto, y saber a cuáles prácticas se les deben dar una mayor prioridad y saber cuáles son de mayor interés para el equipo de desarrollo.

Las 11 prácticas de Martin Fowler son las siguientes:

1. Mantener un único repositorio fuente.
2. Automatizar el *build* (compilación).
3. Hacer el *build* auto-testable.
4. Todos realizan *commits* todos los días.
5. Cada *commit* gatilla el *build* en una máquina de integración.
6. Corregir *builds* fallidos inmediatamente.
7. Mantener el *build* rápido.
8. Ejecutar pruebas en una máquina clon a la del entorno de producción.
9. Todos pueden obtener la última versión estable de la aplicación.
10. Todos pueden ver qué está pasando.
11. Automatizar el *deployment*.

3.4.2. Jenkins

Jenkins²⁹ es el servidor de Integración Continua de código libre que realiza la construcción automática (*build*) del repositorio de MusycGo, ejecuta pruebas y detecta los problemas de compilación o de ejecución de pruebas. Además, cuenta con una extensa variedad de *plugins* para distintas tareas relacionadas con el control de versiones, automatización de *builds*, generación y despliegue de reportes, respaldos de las bases de datos, entre otros.

Ejemplificando de manera simple el proceso que realiza Jenkins en este proyecto es que con cada *commit-push* en la rama principal gatilla en el servidor la ejecución de un *build* y luego se lanzan tareas declaradas previamente, cuando finalizan estas tareas se despliega el *build* en un entorno de producción.

La Figura 24 muestra la ventana principal de Jenkins, se encuentran las cuatro principales tareas que realiza luego de detectar un cambio en el repositorio, estas son: respaldo de la

²⁹ <https://jenkins.io/>

base de datos diariamente, generación de la documentación del código fuente, construcción del proyecto y la tarea que inicializa del servidor del proyecto.

The screenshot shows the Jenkins dashboard with a list of build jobs. The table has columns: S (Status), W (Workstation), Nombre (Name), Último Éxito (Last Success), Último Fallo (Last Failure), and Última Duración (Last Duration). The jobs listed are:

S	W	Nombre	Último Éxito	Último Fallo	Última Duración
		Backup_DB	15 Hor - #239	6 Mes 5 días - #50	53 Ms
		Compodoc	1 Hor 57 Min - #191	2 Mes 29 días - #110	16 Seg
		MusycGo-Project	1 Hor 59 Min - #265	19 días - #244	1 Min 49 Seg
		Server	1 Hor 57 Min - #202	5 Mes 29 días - #18	2.4 Seg

Icons at the bottom: S M L. Buttons: Guía de iconos, RSS para todos, RSS para fallas, RSS para los más recientes.

Figura 24. Servidor de Integración Continua para MusycGo vía Jenkins

3.4.3. Bitbucket

Bitbucket³⁰, lanzado el año 2008 por la empresa “Atlassian”, es un servicio para manejar el versionamiento de código fuente de un proyecto, que registra los cambios realizados sobre un archivo o conjunto de archivos a lo largo del tiempo de tal manera que sea posible recuperar versiones del código más adelante.

Dentro de un repositorio de Bitbucket se encuentra almacenado el proyecto MusycGo (ver Figura 25) utilizando el sistema de control de versiones de Git y a través de un *software* llamado Sourcetree³¹ se gestiona el versionamiento del proyecto.

The screenshot shows the Bitbucket repository 'gucapp_repo'. The sidebar includes links for Source, Commits, Branches, Pull requests, Pipelines, Deployments, Downloads, and Trello. The main area displays the repository details: Last updated 22 hours ago, 0 open pull requests, 4 branches, 2 watchers, 0 forks, Version control system Git, Access level Write, and 0 builds. A message encourages cloning the repository. Below is a file list:

Name	Size	Last commit	Message
documentation		2019-10-04	Documentación de código.
e2e		2019-12-16	Cambios.
graphics		2019-10-10	Url antiguas.
icon		2019-10-10	Url antiguas.
node_modules		2020-01-03	Pestaña de estadísticas. Quitar dependencias que no se ocup...
server		6 days ago	CORS post http debe tener sufijo /api/ según servidor nginx.
src		22 hours ago	Botón para acceder desde el perfil del lugar al perfil maestro ...
.eslintrc.js	440 B	2019-11-12	Eslint. Reporte generado manualmente (eslint.xml).

Figura 25. Versionamiento de código MusycGo vía Bitbucket

³⁰ <https://bitbucket.org/>

³¹ <https://www.sourcetreeapp.com/>

3.4.4. Compodoc

Compodoc³² es una herramienta de código libre con el cual se puede crear la documentación de proyectos llevados a cabo en Angular. Proporciona una documentación clara y útil, mostrando ordenadamente los módulos, clases, dependencias, entre otros.

Compodoc ha sido instalado dentro del servidor de Integración Continua para generar la documentación de manera automática para luego ser desplegada en producción para su fácil acceso desde la web (ver Figura 26).

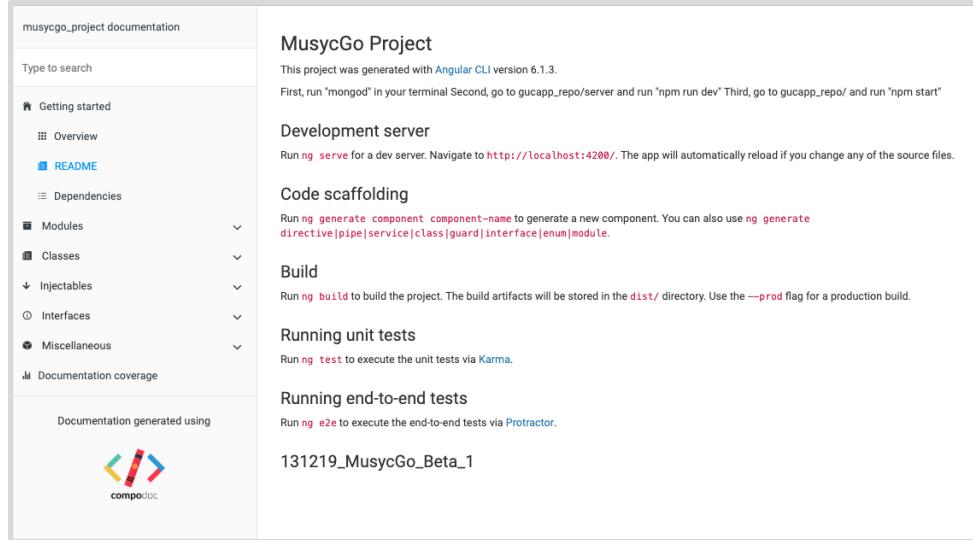


Figura 26. Documentación de código fuente de MusycGo vía Compodoc

3.5. Otras tecnologías utilizadas

Como una manera de aprovechar las tecnologías que existen actualmente, se seleccionaron algunas de ellas como sustento para funcionalidades relevantes del presente proyecto, estas tecnologías se detallan a continuación.

3.5.1. Spotify

El origen de Spotify³³ es sueco y surge de la unión de las dos pasiones de su creador, Daniel Ek, la informática y la música. Junto a Martin Lorentzon, también sueco, crearon Spotify en el año 2006 y fue lanzado en octubre del año 2008.

Para el presente proyecto, la plataforma de Spotify es la biblioteca de música en donde los usuarios pueden obtener los enlaces de las canciones e introducirlas en MusycGo, luego este sistema es el encargado de obtener la información de la canción a través de consultas a la API de Spotify (nombre canción, nombre artista, imagen destacada de la canción,

³² <https://compodoc.app/>

³³ <https://www.spotify.com/>

imagen de artista y género musical), además permite insertar *iframes* de la canción dentro de varios módulos (ver Figura 27).



Figura 27. Ejemplo de *iframe* de Spotify en MusycGo

3.5.2. YouTube

YouTube³⁴ nació en el año 2005, de la mano de Steve Chen, Chad Hurley y Jawed Karim, tres antiguos trabajadores de la empresa “PayPal”, ellos concibieron YouTube como una plataforma digital para alojar contenidos audiovisuales. En el año 2006, fue adquirido por Google por un precio de 1.650 millones de dólares (El País, 2006).

Muchas personas lo usan como una inmensa fuente de conocimiento y por otros como un medio de entretenimiento. YouTube se ha convertido en un medio predilecto en el ecosistema de los medios de comunicación audiovisuales, siendo una gran competencia para la televisión tradicional.

Dentro de MusycGo, YouTube es otro reproductor que se adiciona dentro de la plataforma, insertando el enlace del video en los atributos de una canción es posible reproducir los videos de las canciones ingresadas a través de un *iframe* (ver Figura 28).

³⁴ <https://www.youtube.com/>

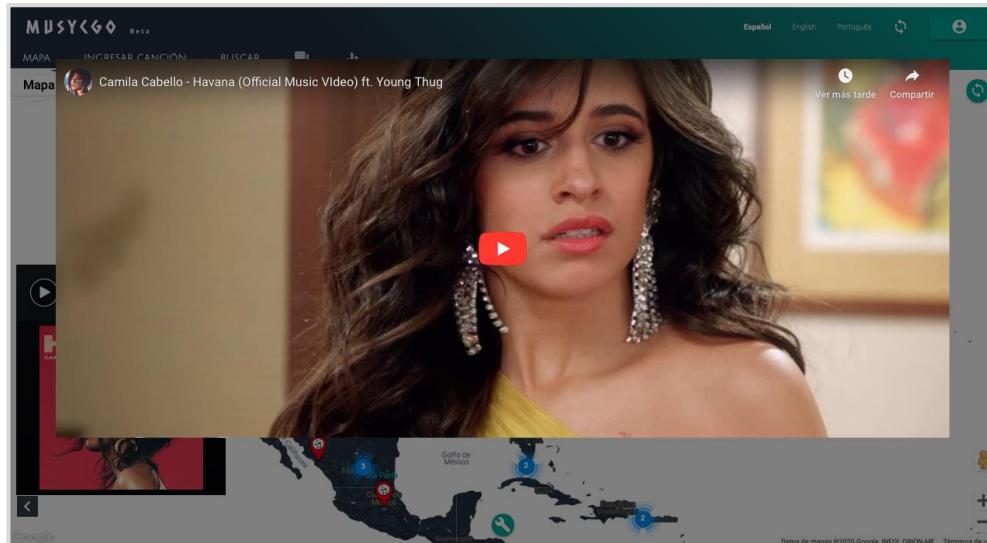


Figura 28. Ejemplo de *iframe* de YouTube en MusycGo

3.5.3. Google Maps

Google Maps³⁵ es un servicio de mapas digitales lanzado principios de 2005 por Google. Ofrece imágenes de mapas desplazables y también imágenes satelitales de cualquier lugar del mundo, incluso muestra las rutas entre diferentes ubicaciones con especificaciones del recorrido en tiempo real.

Esta plataforma ofrece una interacción fluida y amigable para el usuario, permitiendo realizar desplazamientos dentro del mapa, cambiar el nivel de *zoom*, buscar puntos de interés en una zona en particular, compartir una ubicación, obtener información del transporte público, visión en 360° sobre las calles del mundo, entre otras cosas.

Google Maps es el responsable de manejar la interacción del usuario con los mapas dentro de MusycGo (ver Figura 29), y haciendo usos de las funciones que ofrece es posible obtener: coordenadas geográficas, nombre de lugares, desplegar información en marcadores, distancias entre dos puntos, entre otros.

³⁵ <https://www.google.cl/maps/>

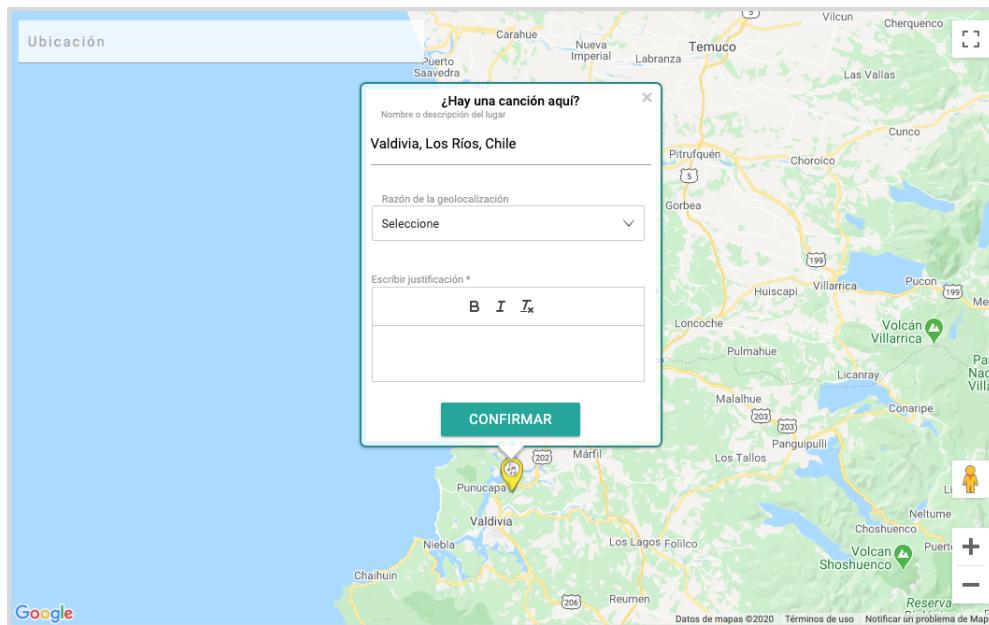


Figura 29. Ejemplo de uso del mapa Google Maps para el ingreso de una canción

3.5.4. Google Analytics

Google Analytics³⁶ es una herramienta gratuita de analítica web de la empresa Google, lanzada en el año 2005.

Este servicio agrupa el tráfico que llega a los sitios web según la audiencia, el comportamiento, plataformas de acceso, entre otros. Es un servicio muy potente porque se puede conocer en tiempo real el número de visitantes, duración de las sesiones, cuáles han sido las URL que han visitado, etc.

Google Analytics se puede considerar como un aliado cuando se pretende establecer una estrategia de negocio, ya que los datos que provee pueden ser valiosos si se analizan e interpretan correctamente, al fin y al cabo, es una manera de conocer a los usuarios y saber qué es lo que quieren y lo que les gusta.

Para MusycGo, este servicio ofrece información en tiempo real del número de usuarios activos, estadísticas de páginas más visitadas dentro de la aplicación, descripción básica del tipo de usuario, facilita datos e informes sobre todo lo que pasa en el sitio web (ver Figura 30).

³⁶ <https://analytics.google.com/>

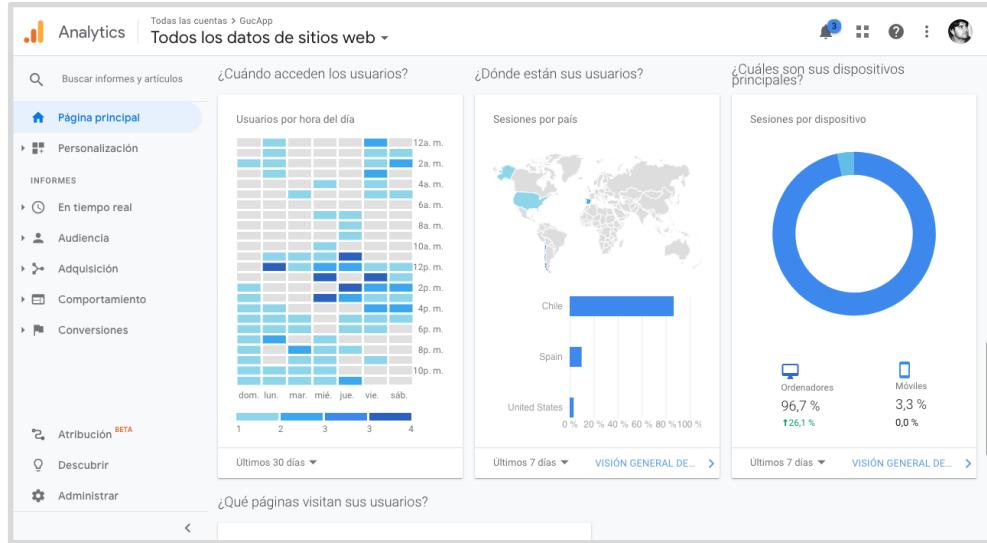


Figura 30. Algunas estadísticas de MusycGo vía Google Analytics

3.5.5. UptimeRobot

UptimeRobot³⁷ es un servicio web que monitorea continuamente los sitios definidos según su URL o IP³⁸. Se verifica cada cierto tiempo si el sitio se encuentra caído o sin respuesta, notificando por correo electrónico cuando esto sucede.

Este servicio actualmente está configurado para monitorear el estado de la URL de MusycGo y la IP del servidor de Integración Continua (ver Figura 31) en intervalos de cinco minutos (valor mínimo en la versión gratuita).

Además, ofrece un historial de los últimos eventos (“*up*”, “*down*”, “*start*” y “*pause*”) para los monitores del proyecto. También muestra un gráfico con los tiempos de respuesta del servidor en las últimas 24 horas.

³⁷ <https://uptimerobot.com/>

³⁸ IP de siglas en inglés *Internet Protocol* (Protocolo de Internet).

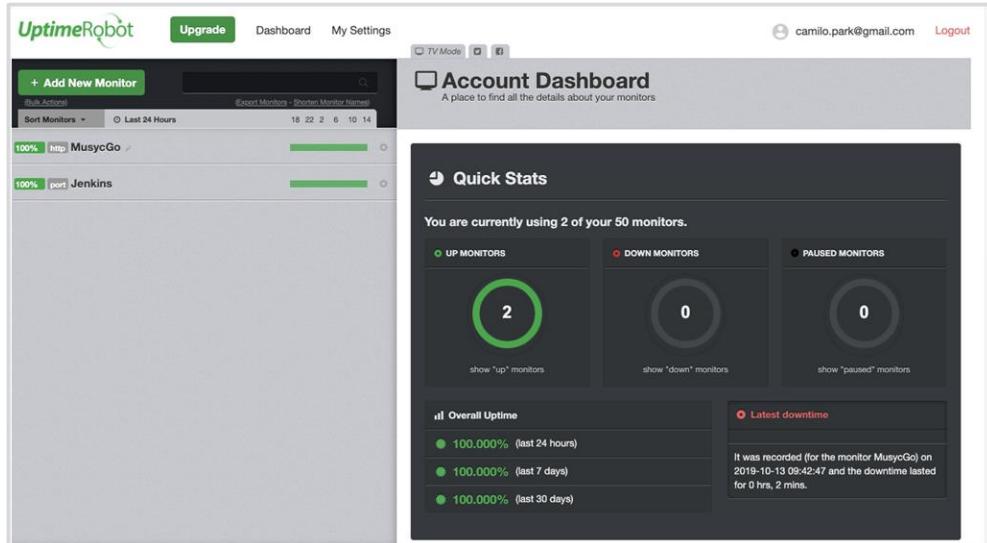


Figura 31. Monitorización del estado de MusycGo vía UptimeRobot

3.5.6. Disqus

Disqus³⁹, fundada en 2007 por Daniel Ha y Json Yanes, es un servicio en línea de gestión de comentarios que se puede insertar en sitios web.

Este servicio incluye características como: integración de redes sociales, perfiles de usuario, herramientas de *spam* y moderación, análisis de datos, posibilidad de reaccionar a la página a través de emoticones y notificaciones por correo electrónico.

Disqus permite dejar comentarios solo con un registro previo ingresando las credenciales de Facebook, Twitter, Google, OpenID o Yahoo.

La plataforma de Disqus ofrece una robusta forma de gestionar los comentarios de los usuarios para todas las canciones ingresadas en MusycGo (ver Figura 32), con el cual se puede iniciar hilos de comentarios anidados, valorarlos y poder recibir correos electrónicos cuando se produzca una nueva interacción.

³⁹ <https://disqus.com/>



Figura 32. Ejemplo de uso de Disqus para el ingreso de comentarios en MusycGo

3.5.7. Auth0

Fundada en el año 2013 por Eugenio Pace y Matías Woloski, Auth0⁴⁰ es un servicio que se encarga de implementar funciones relacionadas con la autenticación para una aplicación o sitio web, es decir, es un servicio que permite externalizar las funciones para los registros de usuarios e inicio de sesiones.

Algunas de las principales características que tiene son: sencillo inicio de sesión (a través de Facebook, Twitter, Gmail, entre otros), autenticación multi-factor, inicio de sesión sin contraseña, gestión de usuarios, etc.

En MusycGo se externaliza el proceso de registro e inicio de sesión a Auth0 (ver Figura 33). MusycGo a través de una devolución de llamada capta la información del usuario autenticado, lo que permite realizar acciones pertinentes para almacenar los datos en la base de datos propia de MusycGo.

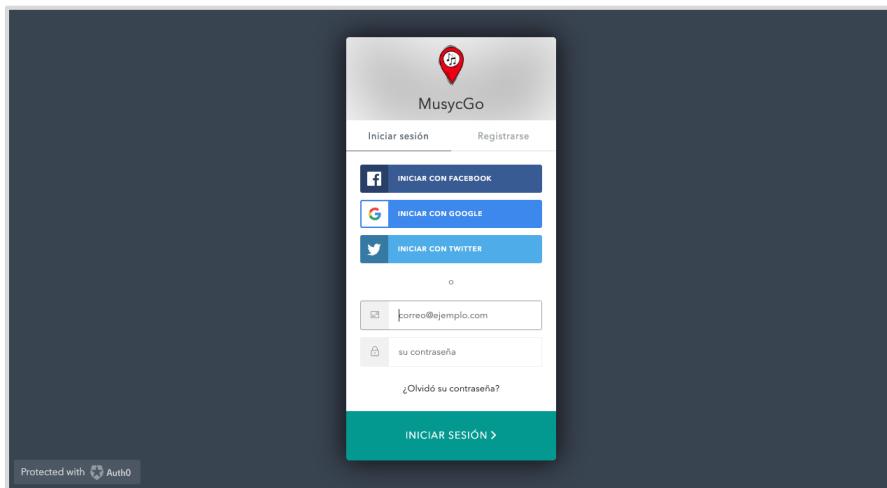


Figura 33. Registro e inicio de sesión en MusycGo vía Auth0

⁴⁰ <https://auth0.com/>

3.6. Diseño de navegación

La Figura 34, muestra el diagrama de navegación que describe todas las vistas de MusycGo y cómo interactúan para acceder de una vista a otra.

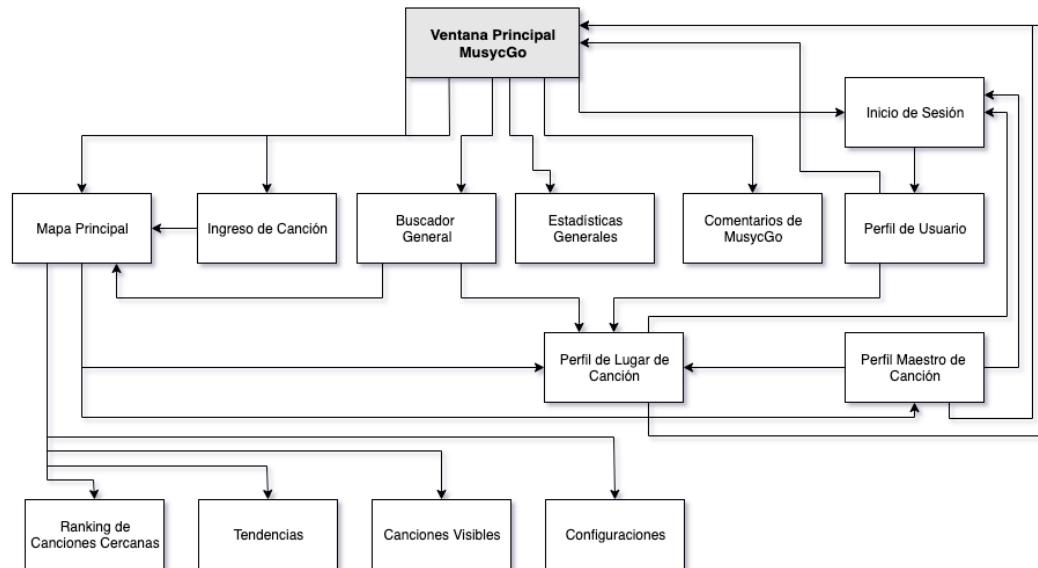


Figura 34. Diagrama de navegación MusycGo

3.6.1. Ventana principal MusycGo

Es la primera vista con la que se encuentra el usuario cuando ingresa a la aplicación. Al ser la ventana principal, se puede acceder desde aquí a las demás vistas del sistema.

Está constituida de dos partes, la primera es el encabezado del sitio, donde se muestra: el nombre de la aplicación, cinco pestañas para acceder a otras vistas, opciones de los idiomas disponibles (español, inglés y portugués), un botón para acceder a las redes sociales, otro botón desplegable para acceder a información de MusycGo y al formulario de contacto con el equipo de MusycGo, un botón para ver las notificaciones, otro para recargar la página y finalmente el botón de acceso a la autenticación. Mientras que la segunda parte es el mapa principal, ocupando la mayor parte de la pantalla (ver Figura 35).

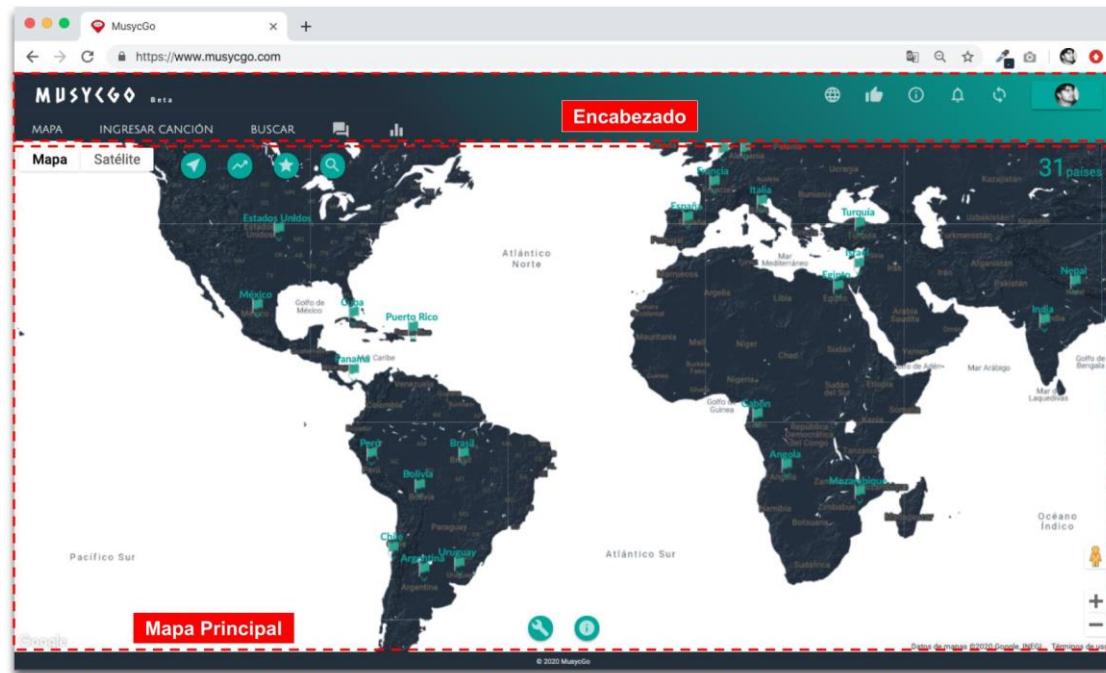


Figura 35. Ventana principal MusycGo

3.6.2. Mapa principal

Vista que corresponde a la primera pestaña (de izquierda a derecha) del encabezado de la ventana principal. Este mapa muestra todas las canciones georreferenciadas. Presenta diferentes modos de interacción y configuración para mostrar y desplegar información.

Cabe señalar de antemano que cada botón de la página principal cuenta con una breve descripción de su funcionalidad, esto se muestra cuando se posiciona el cursor sobre el elemento. Estos mensajes también se pueden ver en otras páginas.

Mostrando la Figura 36 como referencia, las principales funcionalidades de esta vista son:

- **Sección 1:** Botones para cambiar el estilo del mapa entre: “Mapa” y “Satélite”.
- **Sección 2:** Botones con funcionalidades dentro del mapa, las cuales son (de izquierda a derecha):
 - Acceso a posición GPS del usuario.
 - Ver el *ranking* de las canciones cercanas (requiere que el GPS esté activado).
 - Ver listado de las canciones visibles en el mapa.
 - Ver *ranking* de las tendencias en las reproducciones externas (en Spotify) y las tendencias en las visualizaciones del “perfil del lugar de la canción”.
 - Botón que muestran dos cuadros de texto para realizar búsquedas por ubicación o título de la canción.
 - Botón para cambiar el *zoom* al estado inicial.

- **Sección 3:** Número que indica el número de canciones visibles mientras navega por diferentes lugares desde el mapa.
- **Sección 4:** Botón para actualizar los marcadores en el mapa, en caso de que no se visualicen correctamente.
- **Sección 5:** Botón de interacción para acceder a Street View de Google Maps.
- **Sección 6:** Botones para modificar el nivel de *zoom* en el mapa.
- **Sección 7:** Botón para acceder a la información de la simbología del mapa.
- **Sección 8:** Botón para acceder a la configuración del mapa, tanto a nivel de colores, como de filtro para los datos mostrados.
- **Sección 9:** Botón que permite mostrar u ocultar el reproductor de la canción seleccionada.

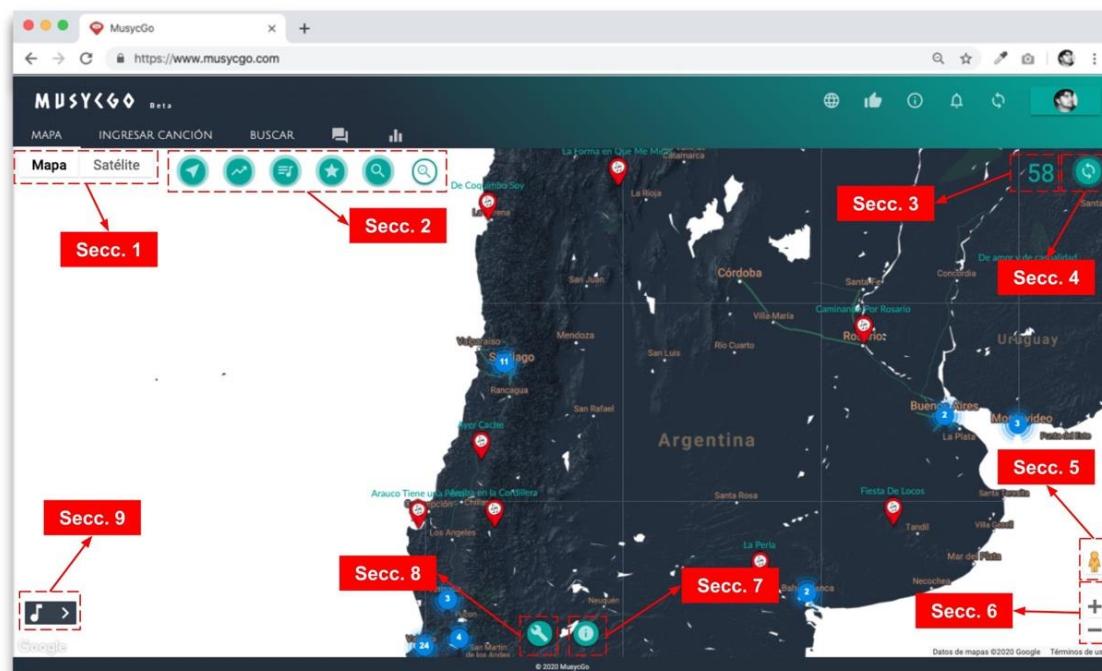


Figura 36. Ventana mapa principal de MusycGo

Otras funcionalidades que ofrece este mapa son:

- Mostrar las colaboraciones de los usuarios en cualquier lugar.
- Agrupar en *clusters* los marcadores cercanos, indicando el número de canciones en cada agrupación.
- Acceder a información de la canción y georreferenciación (ver Figura 37).
- A través de los marcadores se puede: compartir la canción, reproducir (audio y video), buscar las letras de la canción en Musixmatch⁴¹ y agregar la canción a favoritos (ver Figura 37).

⁴¹ <https://www.musixmatch.com/>

- Con el evento de clic sobre cualquier lugar del mapa, se puede acceder a listas de las canciones cercanas a ese lugar y también a la lista de las canciones que se encuentran en el país del lugar, incluyendo un reproductor embebido de la canción en Spotify. Además, para cada elemento de la lista existe un botón para redirigir a los perfiles de las canciones (ver Figura 38).

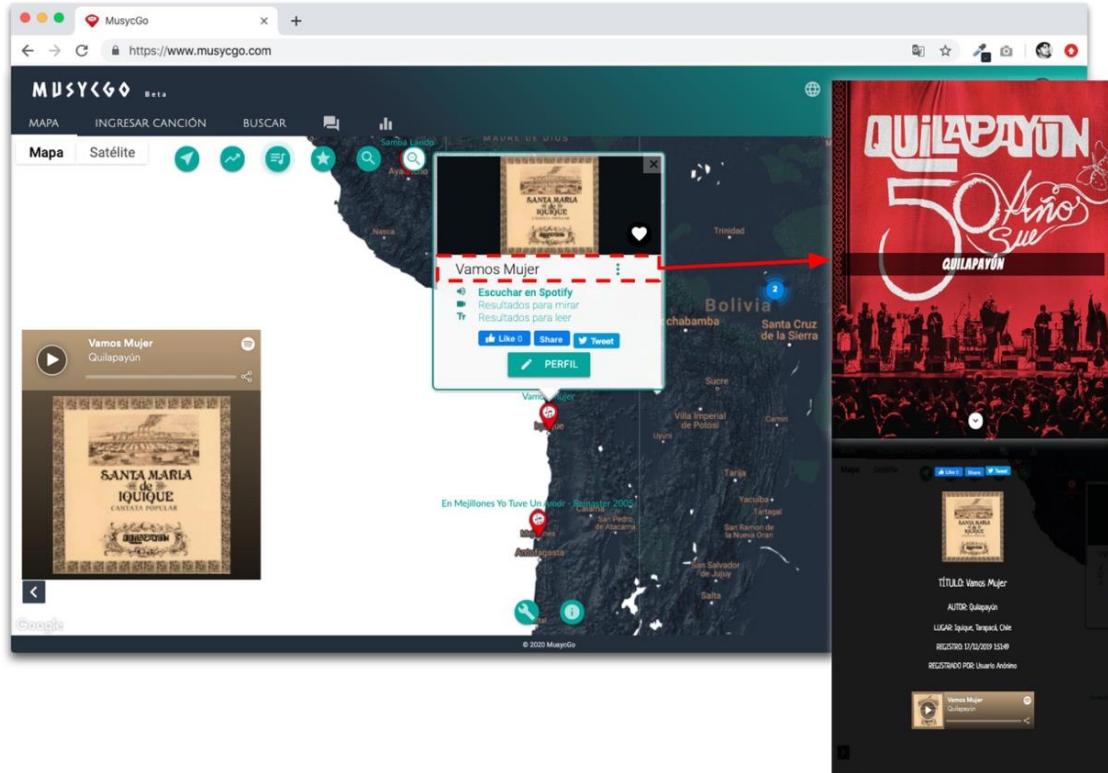


Figura 37. Marcador mapa principal e información extra



Figura 38. Lista de canciones cercanas y por país

3.6.3. Ranking de canciones cercanas

Esta vista es accedida a través de los botones del mapa principal (ver Figura 36, sección 2), y permite mostrar un listado de las canciones cercanas a la posición del usuario (con GPS previamente activado), según un radio de alcance variable. Para cada elemento se muestra la distancia (metros) a la que se encuentra el usuario de la canción. La información que se muestra para cada canción es: nombre, artista, posición en coordenadas de latitud y longitud, la fecha de registro, reproductor embebido de la canción en Spotify y un botón (“Ver en mapa”) que posiciona al usuario en el mapa, precisamente en el marcador de la canción (ver Figura 39).

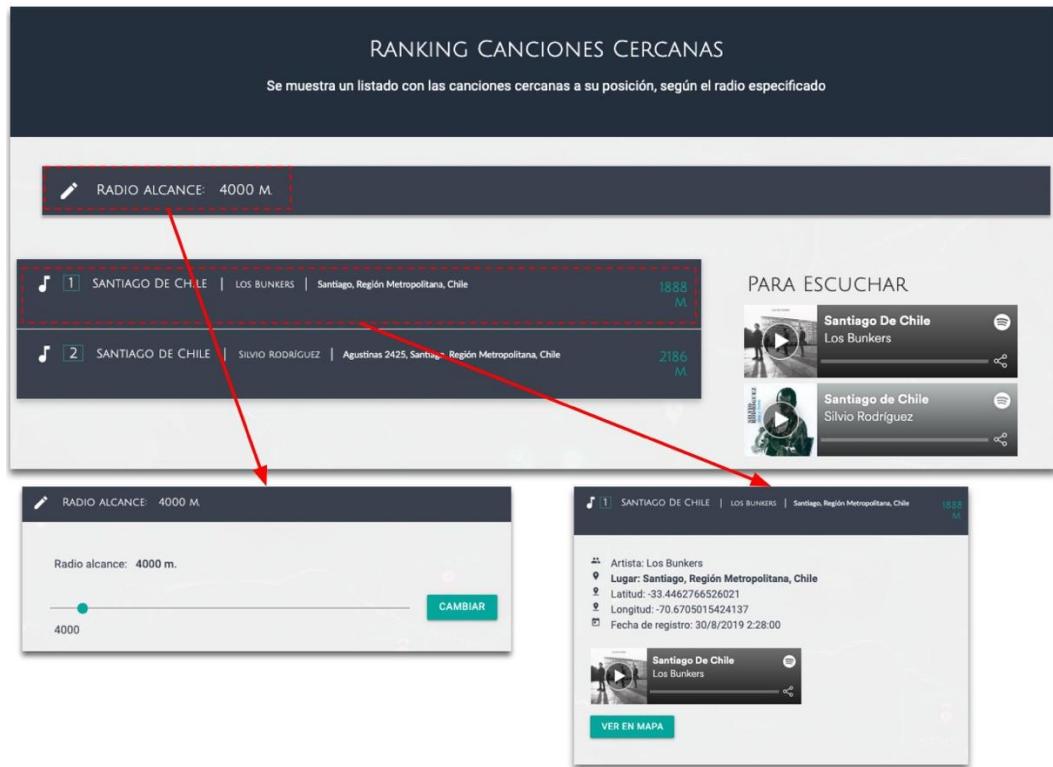


Figura 39. *Ranking* de canciones cercanas, radio de alcance e información extra

3.6.4. Tendencias

Vista que es accedida a través de los botones del mapa principal (ver Figura 36, sección 2), muestra dos listados con las tendencias en reproducciones y visualizaciones.

Como se puede ver en la Figura 40, primero se muestra el listado de las cinco canciones más reproducidas en Spotify (externamente), luego se muestra el listado de las cinco canciones más visualizadas (acceso al “perfil del lugar de la canción”).

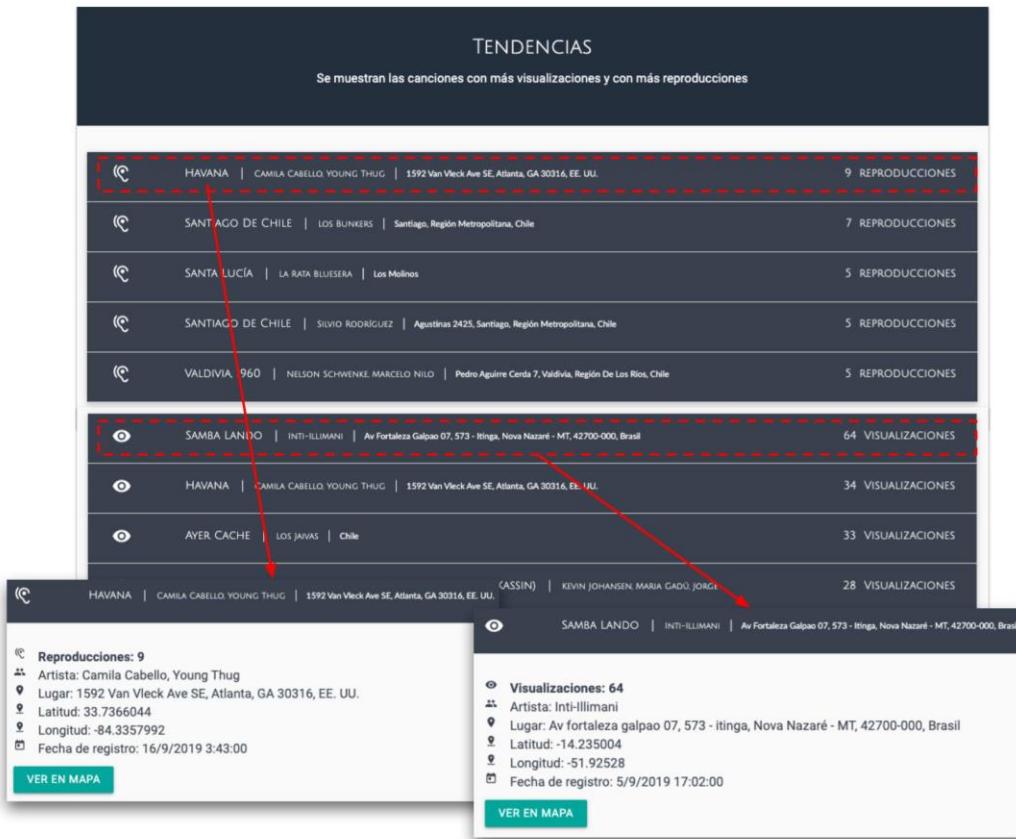


Figura 40. Tendencias en reproducciones y visualizaciones.

Las listas se muestran en orden descendente, cuya información que proporciona cada elemento es: nombre de la canción, nombre del artista, lugar, número de reproducciones o visualizaciones, coordenadas de latitud y longitud, fecha de registro y el botón para ver el marcador en el mapa.

3.6.5. Canciones visibles

Esta vista es accedida a través de los botones del mapa principal (ver Figura 36, sección 2), muestra un listado de las canciones que se encuentran visibles en el mapa, esta lista va cambiando en la medida que el usuario se desplaza por el mapa.

Cada elemento de la lista muestra la siguiente información: nombre de la canción, nombre del artista, lugar, coordenadas de latitud y longitud, fecha de registro y el botón para ver el marcador en el mapa (ver Figura 41).

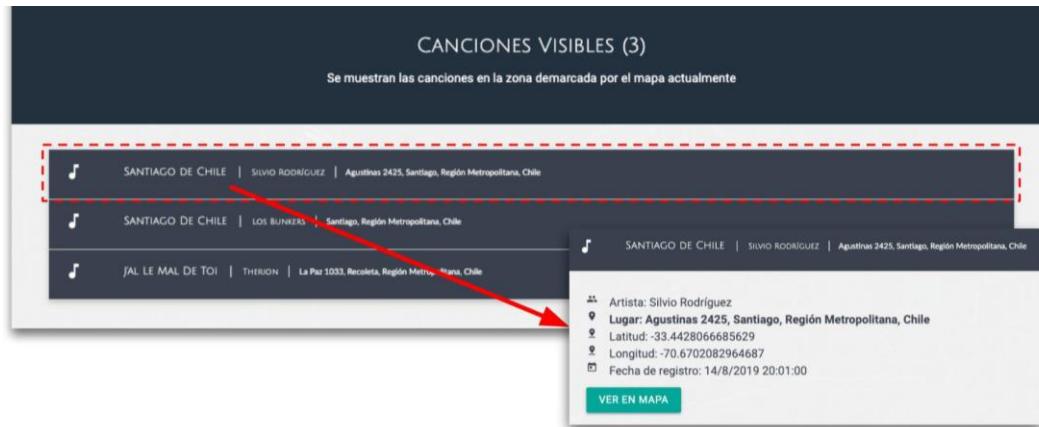


Figura 41. Canciones visibles e información extra

3.6.6. Configuraciones

Esta vista es accedida a través de los botones del mapa principal (ver Figura 36, sección 8), aquí se pueden configurar ciertos parámetros del mapa. También es posible seleccionar un nuevo estilo (colores), además se pueden filtrar los marcadores que son mostrados en el mapa, ya sea por: artista, género musical y por idioma de la letra de la canción (ver Figura 42).

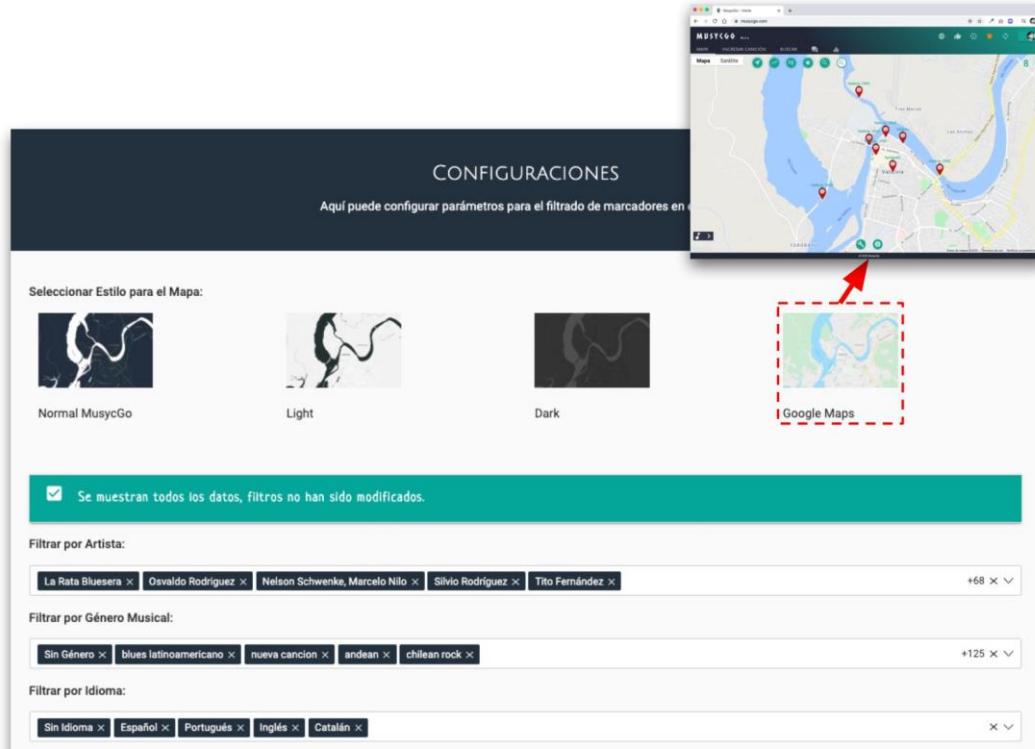


Figura 42. Configuración de mapa, filtros y cambio de estilo

3.6.7. Ingreso de canción

Vista que corresponde a la segunda pestaña (de izquierda a derecha) del encabezado de la ventana principal (ver Figura 35). Existen dos modalidades de ingreso (ver Figura 43).

- **Modalidad 1, Ingreso por enlace de Spotify:** Es la manera que requiere menos datos por parte del usuario, para ingresar una canción se necesita: el enlace de la canción en la plataforma de Spotify (se puede acceder a un pequeño tutorial de cómo obtener este enlace), coordenadas en latitud y longitud del lugar (clic en el mapa), nombre del lugar, la justificación del por qué la canción debe estar en ese lugar y opcionalmente se puede ingresar la letra la canción.
- **Modalidad 2, Ingreso por formulario:** Esta manera requiere más datos que la modalidad 1, ya que necesita: nombre de la canción, nombre del artista, latitud y longitud del lugar (clic en el mapa), nombre del lugar, justificación de la elección del lugar, opcionalmente se puede proporcionar: el género de la canción, el enlace de la canción en Spotify, el enlace del video en YouTube y el enlace de la canción en SoundCloud.

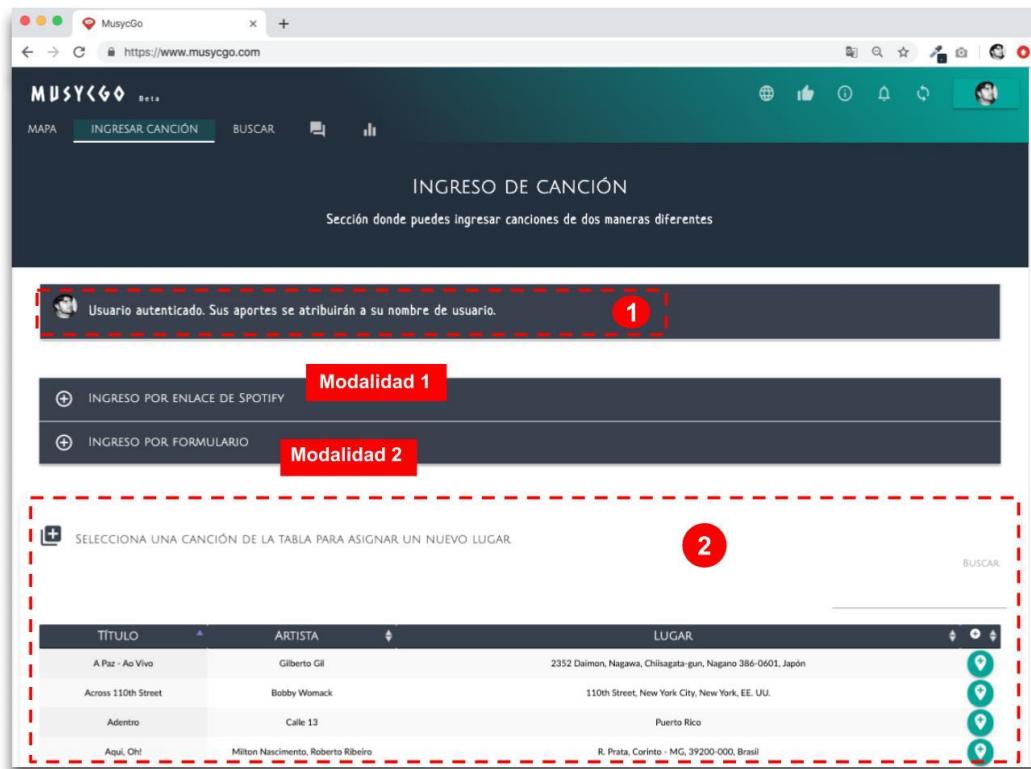


Figura 43. Ventana ingreso de canción

En la parte superior de las dos modalidades de ingreso de canciones se encuentra un recordatorio que indica si el usuario inició sesión o no, mostrando los beneficios de esta acción (ver Figura 43, punto 1).

Dentro de esta vista se pueden ingresar nuevos lugares a canciones ya ingresadas, a través de la selección de una canción en la tabla que se encuentra en la parte inferior (ver Figura 43, punto 2).

3.6.8. Buscador general

Vista que corresponde a la tercera pestaña (de izquierda a derecha) del encabezado de la ventana principal (ver Figura 35). En esta ventana se puede encontrar y visualizar cualquier canción georreferenciada en la aplicación.

Está constituida por una tabla que ofrece las siguientes funcionalidades:

- Buscar una canción georreferenciada (ver Figura 44, punto 1).
- Ordenar (ascendente y descendente) por título, artista, lugar y fecha de registro.
- Visualizar el marcador de la canción en el mapa principal (ver Figura 44, punto 2).
- Mostrar rápidamente la información con la que los usuarios pueden colaborar para que la georreferenciación esté más completa (ver Figura 44, punto 3).
- Acceder al perfil de la canción (ver Figura 44, punto 4).

TÍTULO	ARTISTA	LUGAR	FECHA
A PAZ - AO VIVO	GILBERTO GIL	2352 DAMON, NAGANO, CHISAGATA-GUN, NAGANO 386-0601, JAPÓN	5/9/2019 16:43:00
ACROSS 10TH STREET	BOBBY WOMACK	10TH STREET NEW YORK CITY, NEW YORK, EE. UU.	21/8/2019 17:33:00
ADENTRO	CALLE 13	PUERTO RICO	17/12/2019 2:33:30
AQUÍ, OH!	MILTON NASCIMENTO, ROBERTO RIBEIRO	R. PRATA, CORINTO - MG, 39200-000, BRASIL	18/12/2019 10:11:45
ARAUCAMIA	JUANJO MONTECINOS	RÉGION DE LA ARAUCANIA, CHILE	
ARAUCO TIENE UNA PENA	VIOLETA PARRA	ARAUCO, CHILE	
ARICA	LOS CANTORES	SALVO 885, ARICA, REGIÓN DE ARICA Y PARINACOTA, CHILE	
ARRIBA EN LA CORDILLERA	PATRICIO MANNIS	PASO ATACALCO, ANTÚCAR, PROVINCIA DE CARCHI, ECUADOR	

Figura 44. Ventana de buscador general de colaboraciones

3.6.9. Estadísticas generales

Vista que corresponde a la cuarta pestaña (de izquierda a derecha) del encabezado de la ventana principal (ver Figura 35). Esta ventana muestra gráficamente y a través de listas los siguientes datos:

- El número de colaboraciones por usuario de la aplicación, posicionados según este criterio (ver Figura 45, punto 1).
- Número de canciones que tiene cada país, con *ranking* según este criterio (ver Figura 45, punto 2).
- *Ranking* de las canciones que han sido agregadas más veces a favoritos por los usuarios (ver Figura 45, punto 3).
- Las canciones que tienen dos o más lugares registrados, mostrando gráficamente las más ingresadas (ver Figura 45, punto 4).

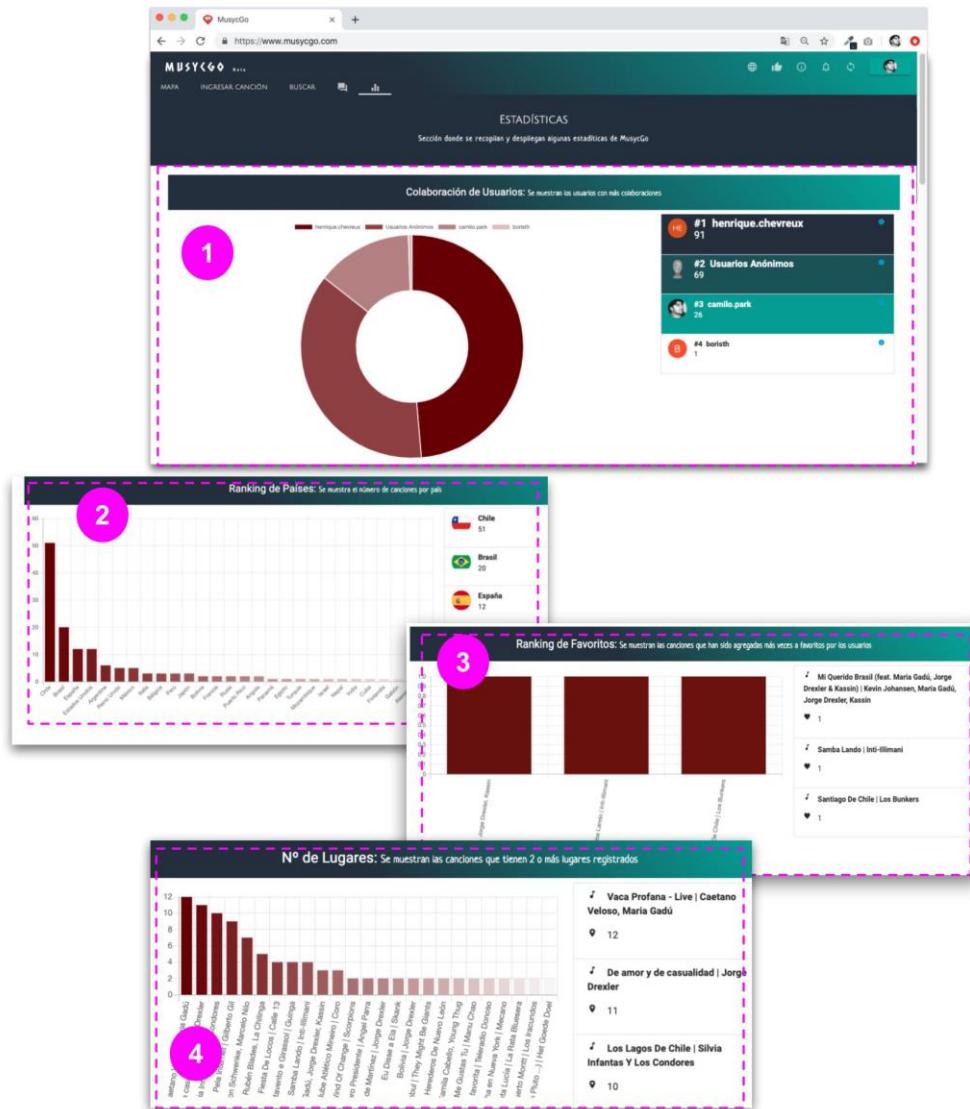


Figura 45. Ventana de estadísticas generales

3.6.10. Comentarios de MusycGo

Se encuentra dentro de la quinta y última pestaña (de izquierda a derecha) del encabezado de la ventana principal (ver Figura 35). En esta ventana se pueden realizar comentarios de la aplicación en general, lo cual permite la interacción de los usuarios (ver Figura 46).

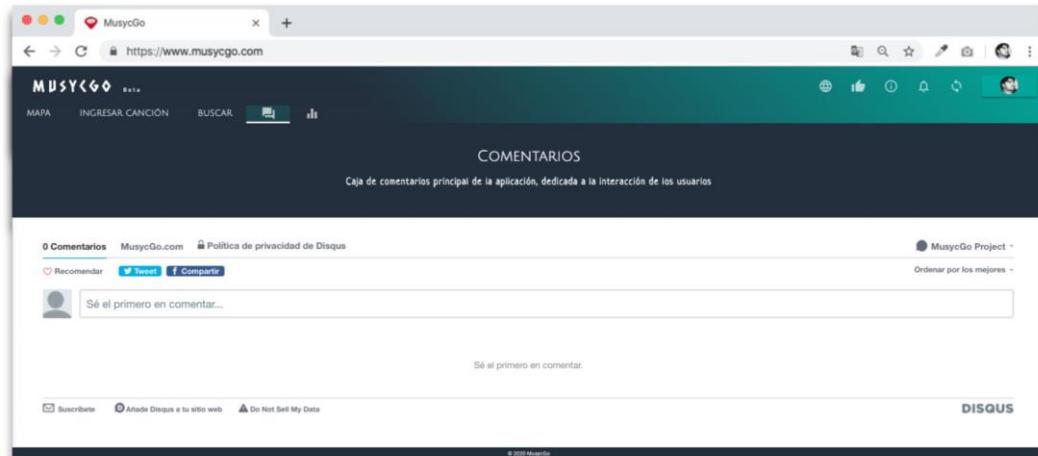


Figura 46. Ventana de comentarios de MusycGo

3.6.11. Inicio de sesión

Es accesible desde la cabecera de la aplicación, al seleccionar el botón de inicio de sesión se produce una redirección a la ventana encargada de esta tarea (ver Figura 47), luego el sistema reconoce al usuario y muestra su información (ver Figura 48).

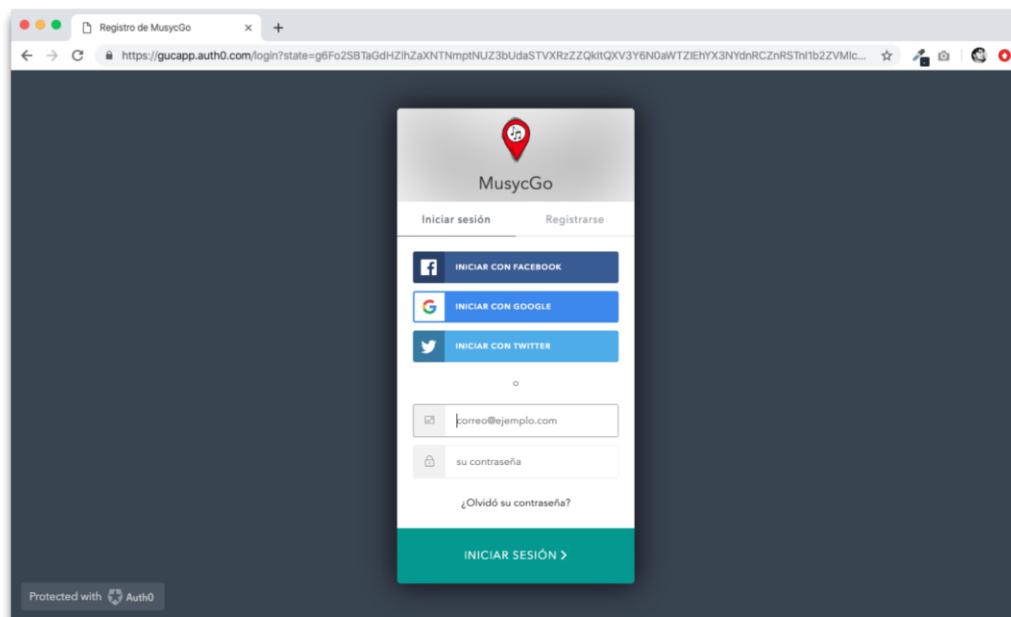


Figura 47. Ventana de registro e inicio de sesión

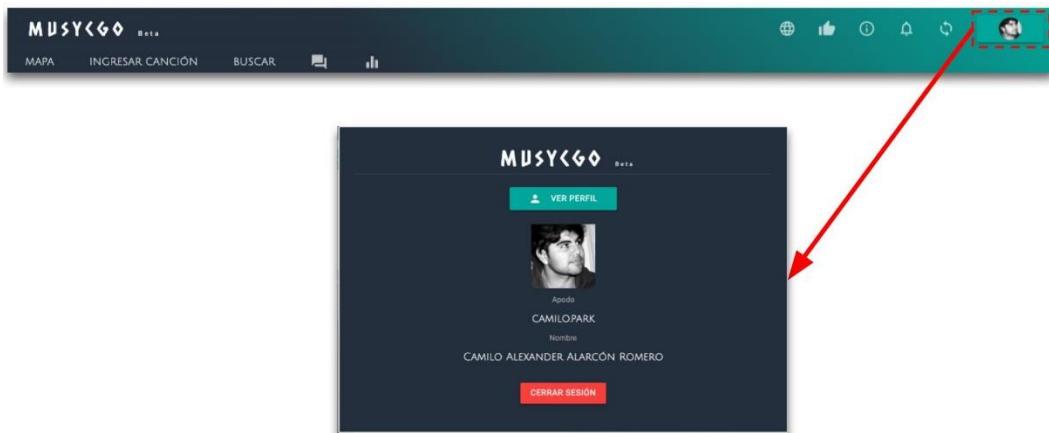


Figura 48. Información del usuario autenticado

Iniciar sesión permite tener características y funcionalidades para los tres tipos de usuarios antes mencionados (usuario explorador, usuario colaborador y usuario moderador), estas funcionalidades son:

- Permitir guardar las canciones en favoritos, a los cuales puede acceder desde el perfil de usuario.
- Permitir registrar las colaboraciones con el nombre de usuario.
- Permitir la edición de las demás canciones registradas en la aplicación.
- Para aquellos usuarios que tengan permisos de moderador, también pueden modificar y/o eliminar canciones georreferenciadas que sean consideradas inadecuadas.

Para el caso en que un usuario haya ingresado un aporte a la aplicación sin haber iniciado sesión previamente, la canción es registrada como usuario anónimo.

Hay que recordar que las vistas en los tres tipos de usuarios es prácticamente la misma, solo que en el usuario moderador se habilitan botones para desvincular un aporte. Por otro lado, se debe mencionar que un usuario explorador pasa a ser considerado colaborador al momento de registrar su primera canción.

3.6.12. Perfil de usuario

Esta ventana muestra la información del usuario autenticado. Primero, se puede ver la información personal, ya sea: imagen del usuario, nombre, apellido y el apodo o *nickname* (ver Figura 49, punto 1).

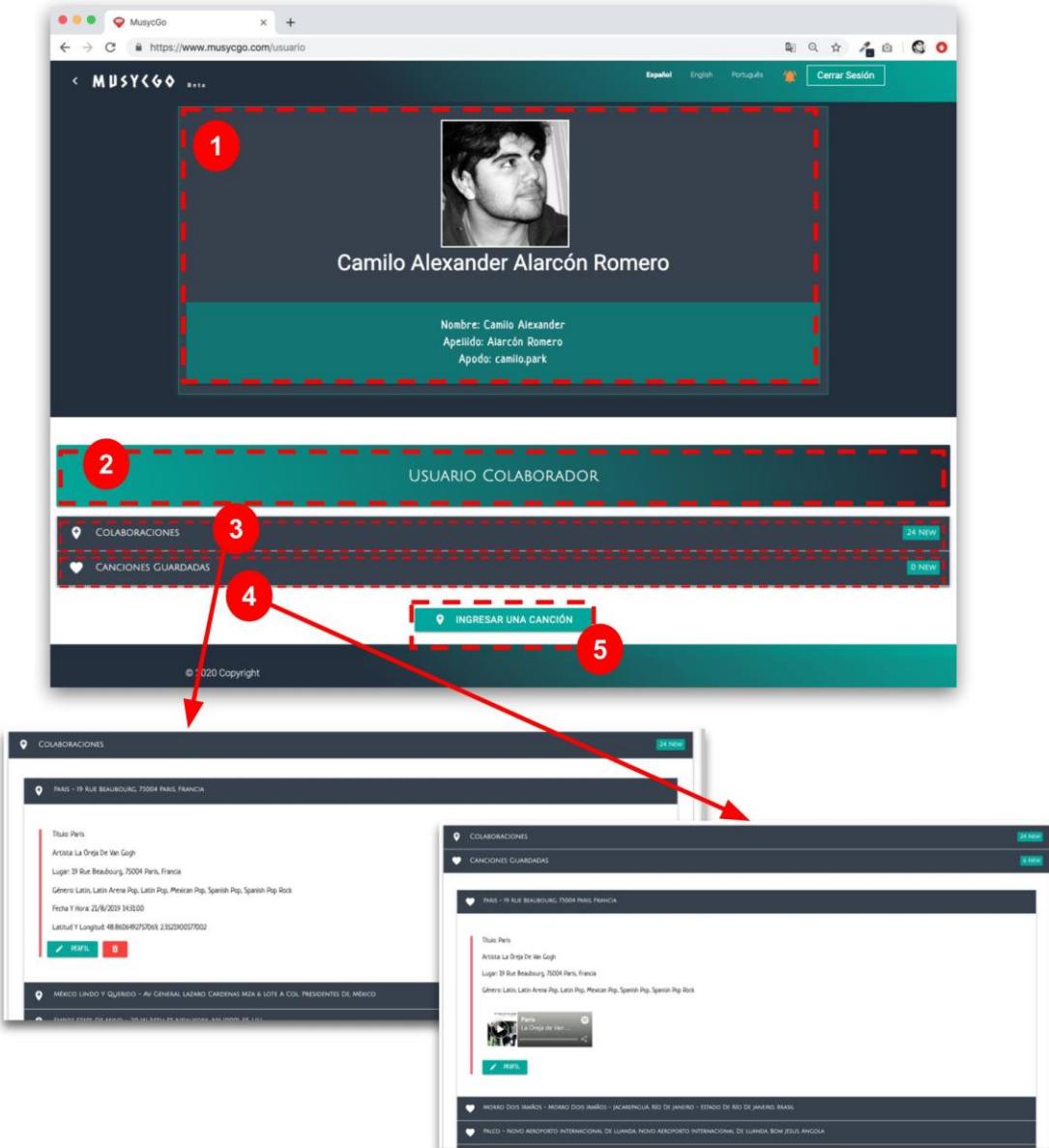


Figura 49. Ventana del perfil de usuario

En la Figura 49, punto 2, se muestra un recuadro que muestra si el usuario se ha comportado como un usuario explorador o colaborador (un usuario no deja de ser explorador cuando es colaborador o viceversa, es decir, se puede ser colaborador y explorador a la vez). Inmediatamente debajo se indican las colaboraciones y las canciones guardadas. Para el caso de las colaboraciones (ver Figura 49, punto 3), al expandir esta sección se puede acceder al listado de todos sus aportes, mostrando: nombre de la canción, artista, lugar, género musical, fecha del registro, latitud y longitud, además existen dos botones que permiten acceder al “perfil del lugar de la canción”, mientras que el otro botón permite eliminar la georreferenciación. Por otro lado, la lista de canciones guardadas (ver Figura 49, punto 4) muestra: número de canciones guardadas, nombre de la canción,

artista, lugar, género musical, un reproductor embebido de la canción en Spotify y finalmente un botón para acceder al “perfil del lugar de la canción”.

También existe un botón que permite redirigir a la ventana principal para poder acceder al ingreso de una nueva canción (ver Figura 49, punto 5).

En la cabecera de la página se muestra el nombre de la aplicación, las opciones para cambiar de idioma, las notificaciones y el botón para cerrar sesión.

3.6.13. Perfil de lugar de la canción

Primero que todo, un lugar de la canción representa una coordenada geográfica diferente a otra, en la cual se añade un texto que señala la dirección aproximada o exacta del lugar que representa dicha latitud y longitud. Hay casos en la que la descripción puede ser más específica que otras, por ejemplo: una canción puede estar inspirada en el “Río Calle-Calle” en Valdivia y un usuario puede optar por escribir como descripción del lugar: “Río Calle-Calle, Valdivia”, “Río Calle-Calle”, “Valdivia” o incluso “Chile”, en cierta forma todas pueden ser válidas, pero hay unas más precisas que otras, es por esto que el sistema permite a la comunidad editar la información, justificar o valorar, para así llegar a la mejor versión.

Esta ventana es diferente para cada lugar de una canción, es el mencionado “Perfil del lugar de la canción”. Por ejemplo, si una canción tiene cinco lugares registrados, existen cinco “perfils del lugar de la canción” diferentes.

Esta vista presenta las siguientes funcionalidades:

- Posibilidad de cambiar de idioma, dentro de los disponibles (ver Figura 50, punto 1).
- Acceder a las notificaciones del usuario (ver Figura 50, punto 2).
- Iniciar sesión, poder acceder al perfil de usuario y la posibilidad de cerrar sesión (ver Figura 50, punto 3).
- Obtener la información principal de la canción: nombre de la canción y nombre del artista, además de ver la imagen destacada de la misma (en caso de no tener una, se muestra la imagen del marcador predeterminado de MusycGo) (ver Figura 50, punto 4).
- Agregar la canción a favoritos y en un costado se muestra el número de usuarios que la han registrado como favorito (ver Figura 50, punto 5).
- Conocer el número de visualizaciones que ha tenido el perfil y también se puede conocer la cantidad de veces que los usuarios han reproducido la canción en Spotify (ver Figura 50, punto 6 y 7).
- Conocer quién fue el usuario que colaboró la canción en el lugar (ver Figura 50, punto 8).
- Conocer la fecha y hora en la que fue realizado el aporte (ver Figura 50, punto 9).
- Compartir el perfil, a través de Facebook o Twitter (ver Figura 50, punto 10).

- Panel multimedia para acceder al audio, video o letra de la canción actual (ver Figura 50, punto 11).
- Información del lugar donde fue registrado (ver Figura 51, punto 1).
- Lista de razones por la que la canción pertenece a ese lugar (ver Figura 51, punto 2).
- Acceder al perfil maestro de la canción (ver Figura 51, punto 3).
- Conocer los demás lugares donde se puede encontrar la canción actual, a través de un mapa y un listado de los lugares (ver Figura 51, punto 4).
- Conocer la justificación y el historial de justificaciones previas, para saber la razón que llevó al usuario a ingresar la canción en ese lugar (ver Figura 51, punto 5 y 6).
- Conocer la letra de la canción (ver Figura 51, punto 7).
- Conocer el género de la canción (ver Figura 52, punto 1).
- Si el usuario está autenticado, puede modificar algunos campos de la canción actual, a través del “Formulario de Modificación”, estos campos son: lugar de la canción, justificación, letra, enlace a YouTube, enlace a SoundCloud y seleccionar idioma(s) de la canción (ver Figura 52, punto 2).
- Los usuarios autenticados pueden calificar la pertenencia de la canción en el lugar, además se muestra el conteo para cada opción de calificación (ver Figura 52, punto 3).
- Caja de comentarios para que los usuarios puedan opinar sobre la georreferenciación (ver Figura 52, punto 4).

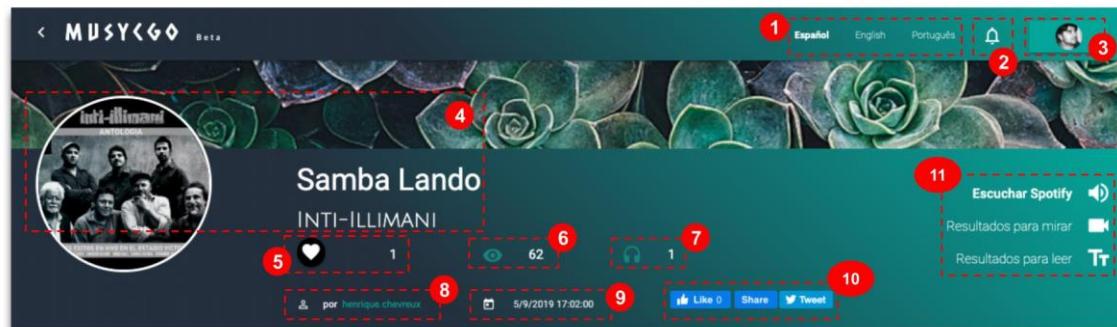


Figura 50. Ventana del perfil del lugar de la canción, parte 1: Encabezado

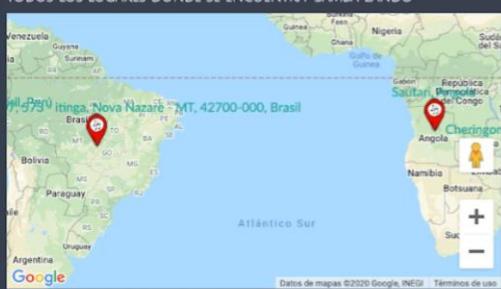
1 

CHERINGOMA, MOZAMBIQUE

2 

MENCIÓN EN LA LETRA

3  VER PERFIL MAESTRO

4 
 TODOS LOS LUGARES DONDE SE ENCUENTRA 'SAMBA LANDO'

- Brasil: Itinga, Nova Nazaré - MT, 42700-000, Brasil
- Mozambique: Cheringoma

 Datos de mapas ©2020 Google, INEGI. Términos de uso

ASOC 16 DE ABRIL, PERÚ
AV FORTALEZA GALPÃO 07 573 - ITINGA NOVA NAZARÉ - MT 42700-000, BRASIL
SAUTARI, ANGOLA
CHERINGOMA, MOZAMBIQUE (LUGAR ACTUAL)

5 JUSTIFICACIÓN

- De Brasil a Mozambique
- Ya no hay nadie que replique

henrique.chevreus (14/12/2019 2:51:00)

6 HISTORIAL DE JUSTIFICACIONES

7 LETRA

Sobre el manto de la noche
 Esta la luna chispeando
 Así brilla fulgurando
 Para establecer un feroz
 "Libertad para los negros
 Cadenas para el negrero"
 Samba landó, samba landó
 ¿Qué tienes tú que no tenga yo?
 Mi padre siendo tan pobre
 Dejo una herencia fastuosa

Figura 51. Ventana del perfil del lugar de la canción, parte 2: Cuerpo

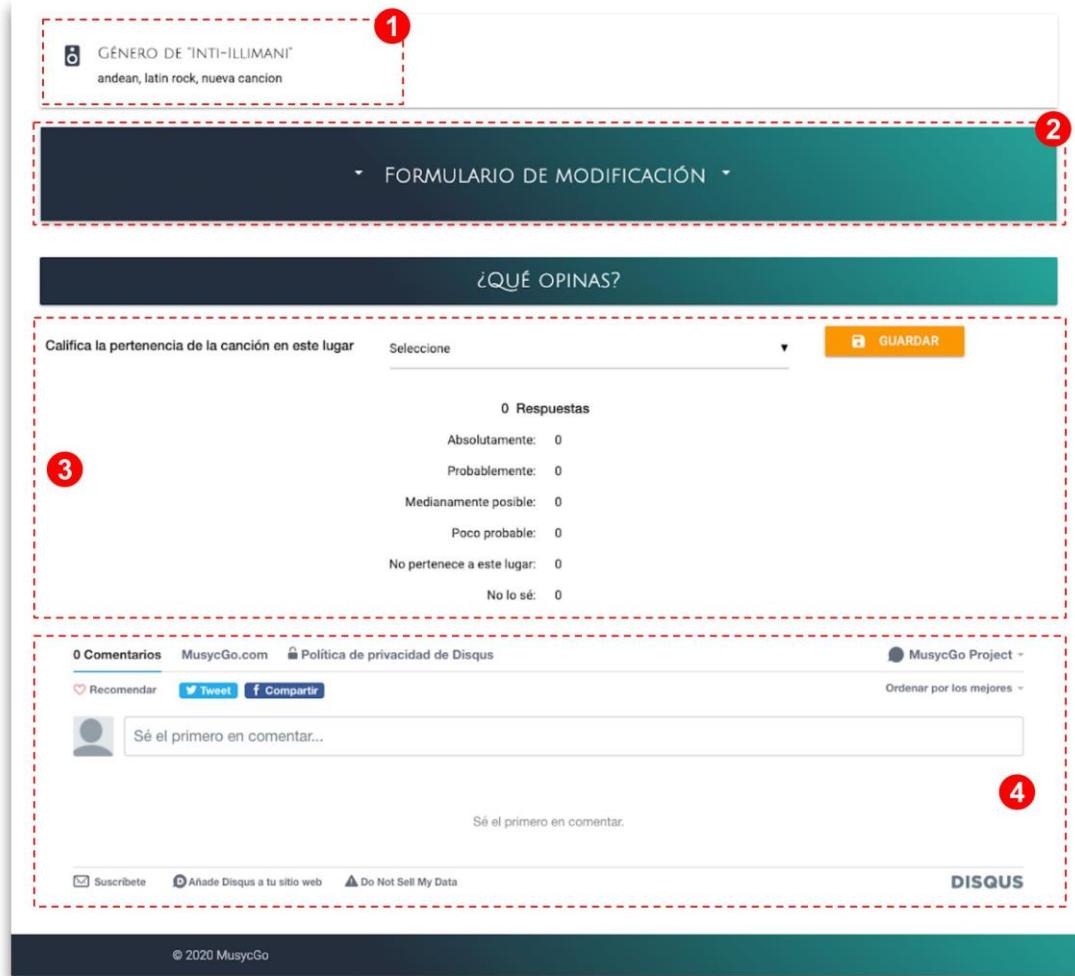


Figura 52. Ventana del perfil del lugar de la canción, parte 3: Cuerpo

3.6.14. Perfil maestro de canción

Esta ventana se encarga de mostrar de forma agrupada todos los lugares donde se encuentra una canción en particular.

Tomando la Figura 53 como referencia se describirán las funciones de esta vista.

- **Sección 1:** Muestra el nombre de la aplicación, los idiomas, botón para ver las notificaciones y otro botón para acceder a las opciones de iniciar sesión, cerrar sesión y ver el perfil del usuario.
- **Sección 2:** Muestra información básica de la canción: una imagen destacada de la música y una imagen destacada del artista, nombre de la canción y el nombre del artista. Existe un botón para agregar la canción a favoritos y otros botones para compartir el perfil en Facebook o Twitter, junto al botón “Like” de Facebook.

Finalmente encontramos el contenido multimedia, en donde se puede ver, escuchar o leer la letra.

- **Sección 3:** Mapa que muestra los lugares donde se encuentra georreferenciada la canción. Debajo del mapa se encuentra la lista de los lugares y para cada elemento se puede obtener la justificación y un botón para acceder al perfil del lugar.
- **Sección 4:** Existe un formulario para modificar los atributos que corresponden a la canción, tales como: letra, enlace del video en YouTube, enlace de la canción en SoundCloud e idiomas de la canción.

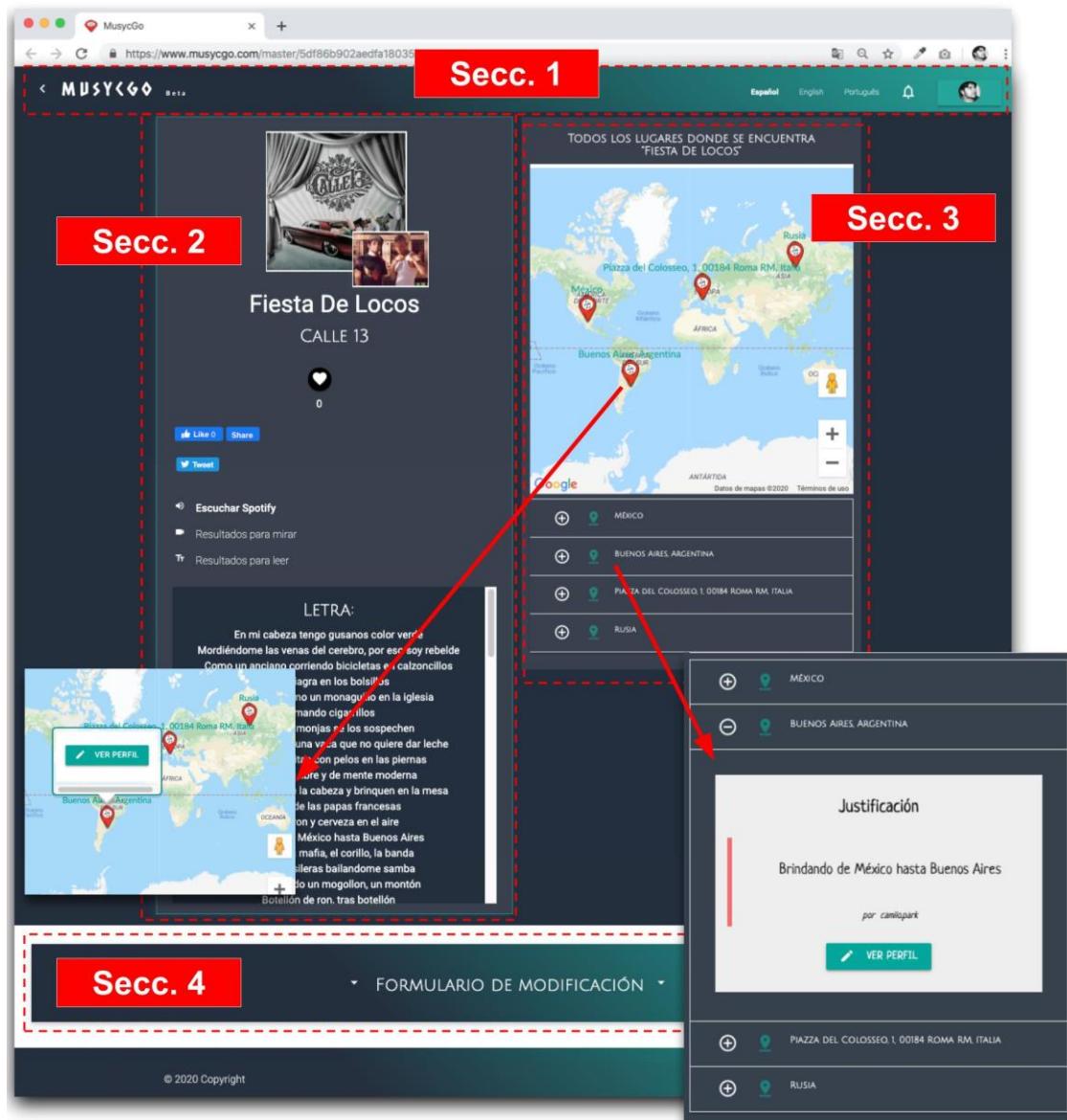
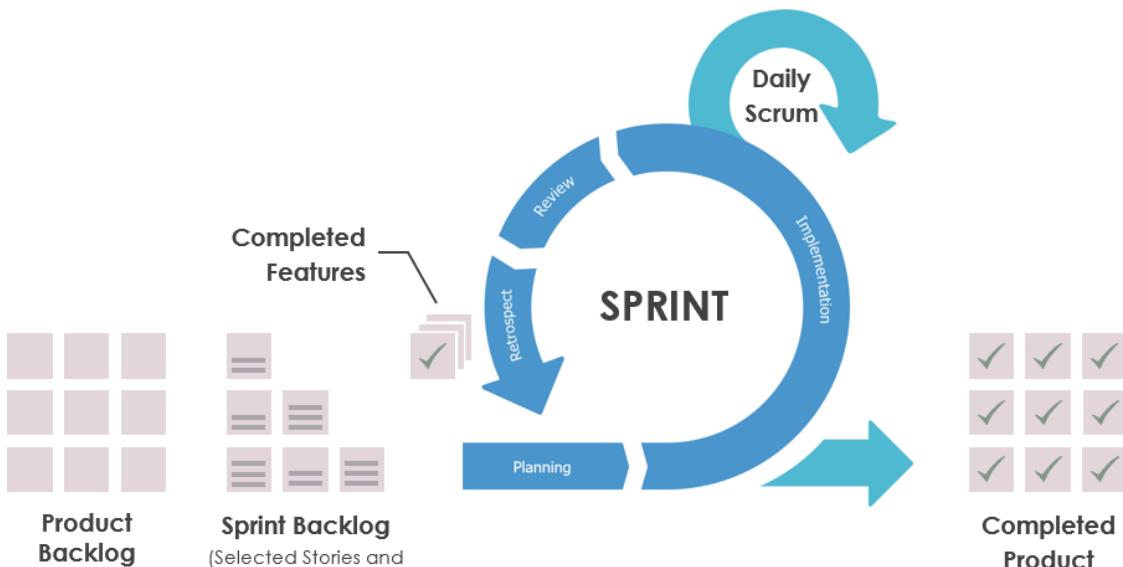


Figura 53. Ventana del perfil maestro de la canción

4. IMPLEMENTACIÓN

4.1. Metodología de trabajo

La metodología con la cual se afrontó y gestionó este proyecto fue la de desarrollo ágil Scrum. Es un marco de trabajo para el desarrollo y el mantenimiento de productos complejos. Se trata de planificar un proyecto en pequeños bloques o *Sprints*, e ir revisando y mejorando el anterior, por lo tanto, es adaptativo (ver Figura 54).



Esta metodología define ciertos artefactos, los cuales se detallan a continuación:

- **Lista de Producto (*Product Backlog*):** Listado de los requisitos escritos como historias de usuario, puede ir cambiando durante el transcurso del proyecto según lo que el producto necesita. Mientras el producto existe, la Lista de Producto también existe.
- **Lista de Pendientes del Sprint (*Sprint Backlog*):** Conjunto de elementos de la Lista de Producto seleccionados para el *Sprint*. Describe las funcionalidades que formarán parte del próximo incremento.

Según (Sutherland & Schwaber, 2013), describe a los Equipos de Scrum como: “Los Equipos Scrum entregan productos de forma iterativa e incremental, maximizando las oportunidades de obtener retroalimentación. Las entregas incrementales de producto “Terminado” aseguran que siempre estará disponible una versión potencialmente útil y funcional del producto.”

Dentro de los roles de un equipo de Scrum (*Scrum Team*) se encuentran:

⁴² https://beertech.ab-inbev.com/blog_detail/2

- **Dueño de Producto (*Product Owner*)**: Responsable de maximizar el valor del producto y del trabajo del Equipo de Desarrollo. Gestiona la Lista de Productos.
- **Equipo de Desarrollo (*Development Team*)**: Profesionales que desarrollan el proyecto.
- **Scrum Master**: Es el líder, facilitador y responsable del Equipo Scrum. Guía al equipo al cumplimiento de la teoría, las prácticas y reglas de Scrum.

En Scrum existen eventos predefinidos para crear oportunidades formales para la inspección y adaptación de algún aspecto, estos eventos son:

- **Sprint**: Iteración de tiempo definido durante el cual se crea un incremento del producto, utilizable y potencialmente desplegable, según lo establecido en la Lista de Pendientes del *Sprint*. Cada nuevo *Sprint* comienza inmediatamente después de terminar el *Sprint* previo.
- **Reunión de Planificación de Sprint (*Sprint Planning Meeting*)**: Reunión donde se planifica el trabajo a realizar durante el *Sprint*.
- **Objetivo de Sprint (*Sprint Goal*)**: Es una meta establecida para el *Sprint* que puede ser alcanzada mediante la implementación de la Lista de Producto.
- **Scrum Diario (*Daily Scrum, Daily Meetings*)**: Reunión diaria con una duración aproximada de 15 minutos para que el Equipo de Desarrollo sincronice sus actividades y cree un plan para las siguientes 24 horas. Se inspecciona el trabajo avanzado desde el último Scrum Diario y se hace una proyección acerca del trabajo que podría completarse antes del siguiente.
- **Revisión de Sprint (*Sprint Review*)**: Se realiza al final del *Sprint* para inspeccionar lo que se hizo y adaptar la Lista de Producto si fuese necesario
- **Retrospectiva de Sprint (*Sprint Retrospective*)**: Oportunidad para que el Equipo Scrum se inspeccione a sí mismo y cree un plan de mejoras que sean abordadas durante el siguiente *Sprint*. Se realiza después de la Revisión de *Sprint* y antes de la siguiente Reunión de Planificación de *Sprint*.

En el presente proyecto de desarrollo se adaptaron los roles para poder desempeñar la metodología Scrum. El profesor patrocinante, Henrique Chevreux Oliveira, desempeñó los roles de *Scrum Master* y Dueño de Producto, además de velar por el cumplimiento de los *Sprint* y del íntegro desarrollo de la Lista de Producto. Mientras que el Equipo de Desarrollo se conformó únicamente por Camilo Alarcón Romero, aspirante al título de Ingeniero Civil en Informática.

La duración de cada *Sprint* variaba según la cantidad y dificultad de los requisitos de la Lista de Pendientes del *Sprint*, que no fue menos de una semana o más de un mes. Cabe señalar que los requisitos Lista de Producto y Lista de Pendientes del *Sprint* fueron incorporados como historias de usuario dentro de la plataforma de administración de proyectos llamada “Trello” (ver Figura 55), esto permitió llevar un seguimiento en tiempo real del estado de cada *Sprint*.

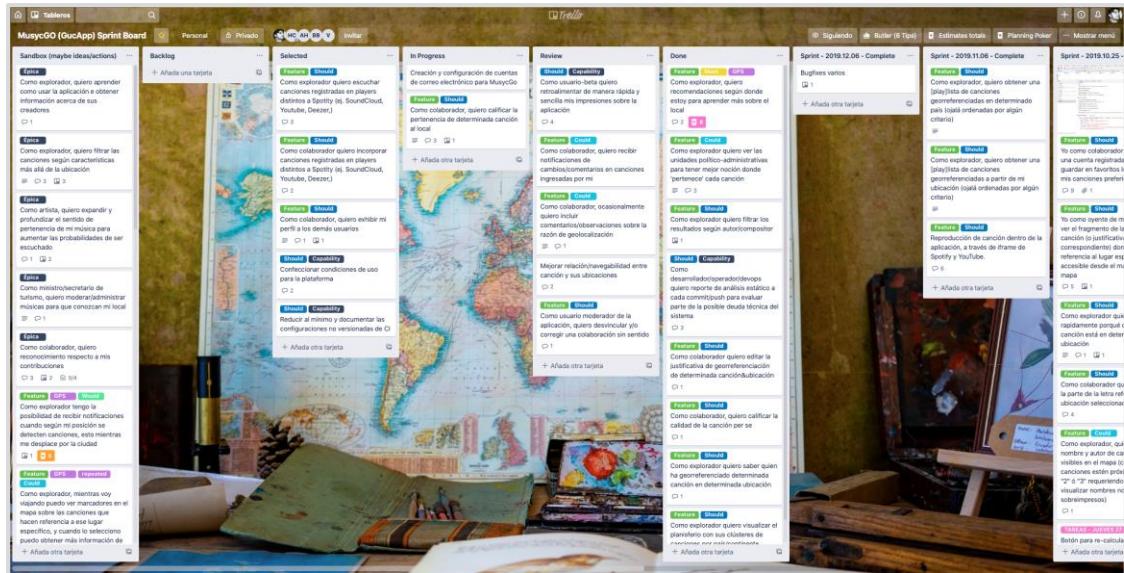


Figura 55. Administración de proyecto MusycGo vía Trello

“Trello es una aplicación basada en el método Kanban y sirve para gestionar tareas permitiendo organizar el trabajo en grupo de forma colaborativa mediante tableros virtuales compuestos de listas de tareas en forma de columnas.” (ENO, 2019).

La existencia de un “tablero” de Trello (*Sprint Board*) con columnas de: “*Backlog*” (Lista de Producto), “*Selected*” (Lista de Pendientes del *Sprint*), “*In Progress*” (En Progreso), “*Review*” (Revisión) y “*Done*” (Terminado), permitió llevar el siguiente proceso: primero se definen los *Sprint* llevando los requisitos seleccionados por el Dueño de Producto desde “*Backlog*” a “*Selected*”, luego el equipo de desarrollo era el encargado de implementar el requisito (moviendo el requisito en desarrollo desde “*Selected*” a “*In Progress*”), cuando la funcionalidad se encontraba lista para ser revisada era movida a “*Review*”, donde el Dueño del Producto realizaba las pruebas de funcionalidad de la historia de usuario y si era aceptada se movía a “*Done*”, caso contrario la historia de usuario era movida a “*In Progress*”. En el momento que se cumplía el plazo definido para el *Sprint* se acordaba una reunión para realizar la Revisión del *Sprint*, Retrospectiva de *Sprint* y finalmente la Reunión de Planificación de *Sprint* siguiente.

En cierto punto del transcurso del proyecto se decidió poner en marcha el Scrum Diario, a través de llamada de voz o video se realizaron breves reuniones de avance para comunicar día a día lo realizado y lo que se realizará. Esto logró un buen entendimiento de las funcionalidades a desarrollar y una excelente comunicación entre el equipo del proyecto.

Por otra parte, cabe mencionar que paralelamente a la metodología Scrum y su tablero virtual para los *Sprint*, existe otro tablero llamado “*BugTracker MusycGo*” destinado a plasmar fallos o “*Bugs*” detectados por el equipo del proyecto y por los usuarios que usan la aplicación. Este tablero cuenta con las columnas: “*Reported by Team*” (Reportado por

el Equipo), “*Reported by Client*” (Reportado por el Cliente), “*Accepted*” (Aceptado), “*In Progress*” (En Progreso), “*To Be Validated*” (Para Ser Validado) y “*Done - Solved*” (Hecho - Resuelto). Los fallos son detectados y luego navegan horizontalmente a través del tablero hasta que son resueltos.

4.2. Diagnóstico y despliegue

4.2.1. Prueba de concepto

Como se mencionó en la sección 3.2 se realizó una prueba de concepto previo al desarrollo del proyecto de título. En aquel concepto se implementó funcionalidades resumidas o incompletas de la idea. El propósito fue verificar que el concepto fuera factible de ser explotado y desarrollado.

A pesar de ser una versión muy básica en un tiempo de un par de semanas, la revisión final y el intercambio de opiniones sobre proceso entre el profesor patrocinante y el tesista, concluyó en que se comenzaría a desarrollar el proyecto hoy llamado MusycGo.

4.2.2. Carpetas

En la Figura 56 se muestra la estructura actual de las carpetas del proyecto MusycGo, se destacan dos carpetas principalmente: “src” y “server”, las cuales corresponden a la carpeta que almacena el código del *frontend* y la carpeta del servidor o *backend* respectivamente.

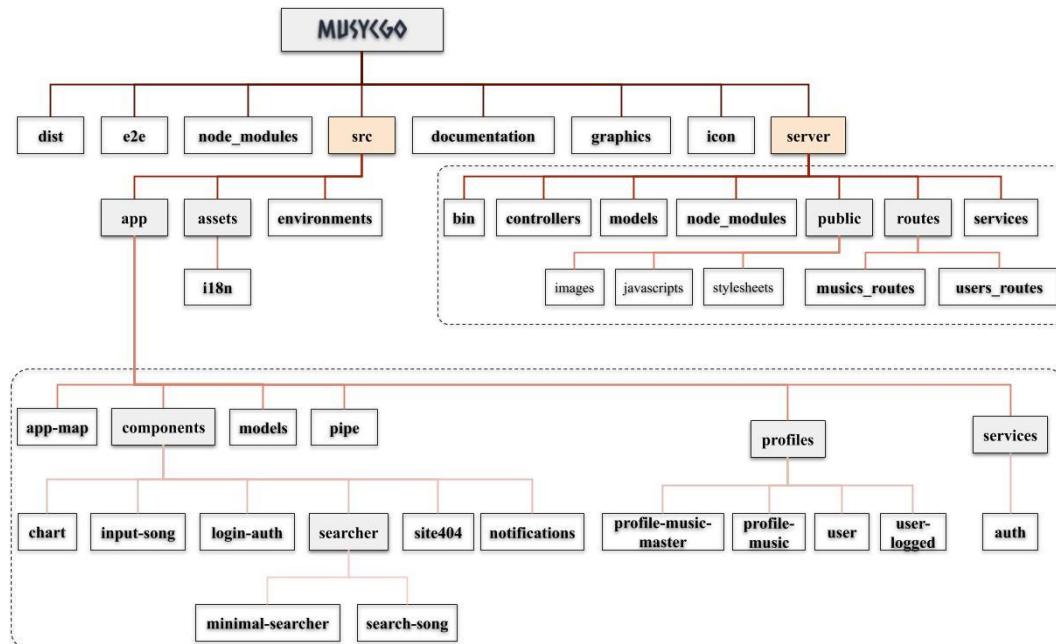


Figura 56. Estructura de carpetas proyecto MusycGo

Recién creado el proyecto de Angular se añadieron automáticamente varias carpetas en la raíz, además de subcarpetas y decenas de archivos que definen el proyecto y configuran el entorno para las diversas herramientas necesarias para el desarrollo.

Las carpetas que fueron generadas en la raíz son:

- **src**: Es donde está ubicado el código fuente del proyecto. Más concretamente en la subcarpeta “app” es donde se programó el proyecto MusycGo. Dentro de la carpeta *source* se encuentran los componentes, servicios, pipes, archivos estáticos, variables de entorno y otros archivos que forman parte de MusycGo. Además, aquí se encuentra el archivo “index.html” que corresponde a la página inicial que es leída por el navegador al momento de ingresar al dominio “www.musycgo.com”.
- **dist**: Aquí se encuentra la versión de la aplicación que estará en el servidor web. Cuando se compila el proyecto los archivos resultantes son llevados a esta carpeta. Hay que mencionar que la compilación se produce en el servidor de Integración Continua (Jenkins), por lo tanto, esta carpeta permanece vacía hasta que se ejecuta la compilación.
- **e2e**: Carpeta que contiene los archivos para realizar pruebas automáticas. Significa “*end to end* testing”.
- **node_modules**: Aquí se encuentran todas las dependencias del proyecto.

Mientras que los archivos que fueron generados en la raíz son:

- **.editorconfig**: Configuración del editor de código.
- **.gitignore**: Almacena las rutas de los archivos que el *software* de control de versiones debe ignorar.
- **angular.json**: Contiene solo la configuración de Angular.
- **package.json**: Contiene la configuración de la aplicación en general.
- **README.md**: Archivo que muestra información de la aplicación.
- **tsconfig.json**: Contiene la configuración de TypeScript.
- **tslint.json**: Archivo que contiene las reglas definidas para analizar el código escrito.

En la Figura 56 se muestran algunas carpetas que fueron generadas manualmente en la carpeta raíz del proyecto, estas son:

- **documentation**: Contiene los archivos generados automáticamente por el documentador de código (Comopodoc). Esta carpeta permanece vacía hasta que el comando para ejecutar la documentación automática es activado por Jenkins.
- **graphics**: Contiene imágenes de tutoriales descriptivos para la aplicación.
- **icon**: Contiene los iconos de los mapas de la aplicación.
- **server**: En esta carpeta se almacena el *backend* de MusycGo. Los archivos son ejecutados en el servidor web para servir los datos a la aplicación.

4.2.3. Despliegue

El despliegue (paso a producción) de la aplicación MusycGo, conllevó a realizar variadas tareas, entre ellas se encuentran:

- Contratación de un plan de *hosting* en Linode⁴³ para alojar el proyecto y que sea visible desde Internet.
- Instalación de servidor web NGINX⁴⁴.
- Contratación del dominio “www.musycgo.com”.
- Contratación del subdominio “api.musycgo.com”.
- Configuración de dominio y subdominio en el servidor.
- Configuración de los DNS⁴⁵ del dominio y subdominio.
- Activación del certificado SSL⁴⁶ en el dominio y subdominio.
- Instalación y configuración de MongoDB en el servidor.
- Instalación y configuración de Jenkins en el servidor.
- Crear tareas en Jenkins (Compilación del proyecto, ejecución servidor virtual, respaldos diarios de la base de datos, generación automática de documentación del proyecto).
- Automatizar la compilación del proyecto en el servidor cuando se detecte un cambio en la rama *master* (o la que esté configurada) del repositorio de MusycGo en Bitbucket.

Actualmente para que el desarrollo en la máquina local pase a producción solo se integra el desarrollo a la rama *master* del repositorio, luego Jenkins ejecuta una serie de tareas automatizadas que concluye en que MusycGo pueda ser accedida desde “www.musycgo.com”.

4.3. Datos

4.3.1. Modelo

El modelo de los datos es representado a través de una base de datos documental (MongoDB), la cual permitió una flexibilidad en el esquema, haciéndolo dinámico por el hecho de que cada registro puede contener información con diferente forma cada vez. También permitió estructuras complejas de datos en un solo documento o registro.

La base de datos que presenta MusycGo almacena la información como un documento, para lo cual se hace uso de una estructura de tipo JSON y que presenta una clave única para cada registro. Esta implementación permite realizar búsquedas por clave-valor, pudiendo realizar consultas avanzadas sobre el contenido del documento.

⁴³ <https://www.linode.com/es/>

⁴⁴ <https://www.nginx.com/>

⁴⁵ <https://www.cloudflare.com/es-la/learning/dns/what-is-dns/>

⁴⁶ <https://www.cloudflare.com/es-la/ssl/>

La base de datos de MusycGo presenta dos colecciones para almacenar los documentos, la primera es la **colección de música** y la segunda es la **colección de usuarios**.

La **colección de música** abarca los datos proporcionados por el usuario al ingresar una canción, los cuales son recopilados por el sistema según los formularios de ingreso o modificación.

Cabe señalar que los datos de la música van cambiando en la medida que el usuario los modifique o por la interacción con algún perfil relacionado. En la Tabla 18 se describe la colección que representa la música en la plataforma.

Tabla 18. Colección de música en la base de datos

Clave	Tipo valor	Obligatorio	Descripción
_id	objectid	Sí	Identificador del documento/registro
_v	int32	No	Clave de versión, describe la revisión interna del documento
artist	string	Sí	Almacena el nombre del artista de la canción
country	array object	No	Arreglo que contiene: latitud, longitud y el nombre abreviado del país donde está ubicada la georreferenciación
date	string	Sí	Fecha y hora en la que fue realizada la georreferenciación
favorite	int32	No	Número de veces que la canción ha sido marcada como favorita por los usuarios
gender	array object	No	Arreglo que almacena todos los géneros musicales del artista y el identificador del artista (recuperados de Spotify)
hear	int32	No	Número de veces que la canción ha sido accedida a ser escuchada en Spotify desde MusycGo
id_user	string	No	Identificador del usuario que realizó la georreferenciación
info	string	No	Campo extra de información
justification	array object	Sí	Arreglo que almacena el historial de justificaciones, contiene los campos: justificación, identificador del usuario, fecha y hora
language	array string	No	Idioma(s) de la letra de la canción
lat	double	Sí	Coordenada de latitud de la georreferenciación
link_soundcloud	string	No	Enlace de la canción en SoundCloud

Tabla 18. (Continuación)

Clave	Tipo valor	Obligatorio	Descripción
link_spotify	String	No	Enlace de la canción en Spotify
link_youtube	String	No	Enlace del video en YouTube
long	double	Sí	Coordenada de longitud de la georreferenciación
look	int32	No	Número de veces que ha sido visitado el “perfil del lugar de la canción”
lyric	string	No	Letra de la canción
name	string	Sí	Nombre de la canción
place	string	Sí	Descripción o nombre del lugar donde se encuentra georreferenciada la canción
reason	array object	Sí	Arreglo que contiene una lista de números que hacen alusión a las razones de la georreferenciación
state	boolean	Sí	Estado de la georreferenciación (<i>True</i> : visible en MusycGo. <i>False</i> : no visible o “eliminada”.)

La **colección de usuarios** abarca los atributos y datos de los usuarios, en su mayoría son obtenidos por el servicio externo Auth0 (sección 3.5.7). Cuando un usuario se “registra” en MusycGo, realmente lo hace en Auth0, este sistema almacena los datos del usuario en su base de datos, luego MusycGo, a través del servicio “auth.service” accede a esos datos en Auth0 y los almacena en una base de datos propia para su posterior reutilización y adhesión de nuevos campos según las necesidades.

A continuación, se presenta la colección que almacena los usuarios de MusycGo (ver Tabla 19).

Tabla 19. Colección de usuarios en la base de datos

Clave	Tipo valor	Obligatorio	Descripción
_id	objectid	Sí	Identificador del documento/registro
_v	int32	No	Clave de versión, describe la revisión interna del documento
contributions	array object	No	Campo extra que almacena las contribuciones de un usuario
date	string	Sí	Fecha y hora del registro del usuario en la plataforma
family_name	string	No	Apellidos del usuario (proporcionado por Auth0)
favorite	array object	No	Arreglo que almacena las canciones favoritas del usuario, contiene los campos: identificador de la canción, y fecha y hora de esta acción
given_name	string	No	Nombres del usuario (proporcionado por Auth0)
name	string	Sí	Concatenación de los nombre y apellidos del usuario (proporcionado por Auth0)
nickname	string	Sí	Nombre de usuario o apodo (proporcionado por Auth0)
notifications	array object	No	Arreglo que almacena el historial de cambios en las georreferenciaciones del usuario, contiene los campos: identificador de la canción modificada, nombre de la canción, artista, lugar, fecha y hora de la modificación, <i>nickname</i> y el campo que verifica si la notificación fue vista o no
picture	string	Sí	Enlace de la imagen del usuario (proporcionado por Auth0)
qualification	array object	No	Arreglo que contiene el identificador de la canción en un lugar específico, acompañado de la opción elegida como calificación de pertenencia para ese lugar
sub	string	Sí	Identificador del usuario (proporcionado por Auth0)

4.3.2. Respaldo

Se aplicaron dos herramientas ofrecidas por el sistema de base de datos (MongoDB), las cuales son “mongodump” y “mongorestore”. Las cuales permiten generar respaldos y restauraciones de los datos de manera local.

Mongodump⁴⁷ es una utilidad para crear una exportación binaria del contenido de la base de datos, esta tarea es ejecutada diariamente por un proceso configurado en el servidor de Integración Continua (Jenkins) contenido en el servidor de MusycGo. Luego, si se

⁴⁷ <https://docs.mongodb.com/manual/reference/program/mongodump/>

requiere una restauración de los datos se ejecuta la utilidad mongorestore⁴⁸ al respaldo de una fecha dada.

4.3.3. Petición de datos

La comunicación del nodo cliente con el servidor se realiza a través de una serie de métodos HTTP. Cuando Angular (nodo cliente) solicita datos al servidor, este último es gestionado por NodeJS, el cual tiene una instancia de ExpressJS, encargado de manejar las rutas de las peticiones HTTP.

Existen dos servicios fundamentales que se conectan directamente con el servidor para el intercambio de datos, estos servicios son: “music.service” y “user.service”. Ambos servicios contienen los métodos para: acceder, insertar, recuperar y modificar los datos de música y usuarios, gestionados a través de diferentes solicitudes POST, GET y PUT. Dentro de todos estos métodos existe la URL de la API a la cual se enviarán las peticiones.

La URL está compuesta por el dominio de la API (“<https://api.musycgo.com>”), seguido del identificador para una ruta de usuario o música (“/api/users” o “/api/musics” respectivamente), finalmente el texto que define a cuál método debe ser redirigido dentro del servidor.

Por ejemplo, se detalla el proceso para obtener los datos de la canción “Santiago de Chile” del grupo “Los Bunkers”:

1. En Angular, un componente llama al método “getMusicById(id: string)” dentro del servicio “music.service”.
2. El método “getMusicById(id: string)” ejecuta una petición GET a la URL: “<http://api.musycgo.com/api/musics/get/by/id/5d68d01adab46962c999a72d>”, siendo el fragmento destacado el identificador de la canción. Luego el método queda en espera de la respuesta del servidor.
3. El servidor recibe una petición y ExpressJS reconoce el fragmento “/api/musics” y dirige la petición al almacén de rutas de la música (“music.routes”).
4. ExpressJS busca el fragmento “/get/by/id” y dirige la petición al controlador de la música (método “music.getMusicById” en el controlador “music.controller”).
5. En el controlador de la música se toma el identificador de la canción (“5d68d01adab46962c999a72d”) y se ejecuta la consulta a la base de datos en MongoDB (método “findById”).
6. Cuando la base de datos retorna un resultado, es empaquetado en una respuesta de tipo JSON.
7. Luego el servidor es el encargado de enviar la respuesta al nodo cliente, cuya respuesta se observa en la Figura 57.

⁴⁸ <https://docs.mongodb.com/manual/reference/program/mongorestore/>

```

1  {
2    "gender": [
3      {"id_artist": "3RTAXX6KGdljBs0IupyZgT",
4       "gender": "chilean rock"
5     }, {
6       "id_artist": "3RTAXX6KGdljBs0IupyZgT",
7       "gender": "concepcion indie"
8     }, {
9       "id_artist": "3RTAXX6KGdljBs0IupyZgT",
10      "gender": "latin alternative"
11    }, {
12      "id_artist": "3RTAXX6KGdljBs0IupyZgT",
13      "gender": "latin rock"
14    }, {
15      "id_artist": "3RTAXX6KGdljBs0IupyZgT",
16      "gender": "nueva cancion"
17    }, {
18      "id_artist": "3RTAXX6KGdljBs0IupyZgT",
19      "gender": "rock en espanol"
20    }],
21    "language": ["-"],
22    "reason": [],
23    "justification": [
24      {"justification": "", "id_user": "NN", "date": "Fri Aug 30 2019 03:28:00 GMT-0300 (GMT-03:00)"}
25    ],
26    "country": [
27      {"name": "CL", "coord": {
28        "lat": -35.675147,
29        "lng": -71.542969
30      }}
31    ],
32    "_id": "5d68d01adab46962c999a72d",
33    "name": "Santiago De Chile",
34    "artist": "Los Bunkers",
35    "lyric": "",
36    "look": 4,
37    "hear": 7,
38    "lat": -33.4462766526021,
39    "long": -70.6705015424137,
40    "place": "Santiago, Región Metropolitana, Chile",
41    "state": true,
42    "info": "",
43    "id_user": "NN",
44    "link_spotify": "https://open.spotify.com/track/3D4izi5jHi4jxG8r4pta7z?si=IhQnjifJTsOfLoWJqwrPnw",
45    "link_youtube": "",
46    "date": "Fri Aug 30 2019 03:28:00 GMT-0300 (GMT-03:00)",
47    "__v": 0,
48    "link_soundcloud": "",
49    "favorite": 1
50  }
51  }
52  }
53  }

```

Figura 57. Respuesta de petición de datos a API MusycGo

Para las peticiones que realizan una inserción, modificación y recuperación de datos, el proceso es casi idéntico, la diferencia radica en la URL en que se accede a la API y en las funciones que manejan estas solicitudes. Siempre son retornados datos o mensajes de estado en formato JSON.

4.4. Componentes de MusycGo

Los componentes constituyen el desarrollo de MusycGo, estos son variados en contexto y funcionalidades para cumplir con los requerimientos planteados para este proyecto.

4.4.1. Mapa

Este componente es visualizado en la pantalla principal de MusycGo (ver Figura 35), el cual corresponde al contenido y funcionalidades dentro de los límites del mapa.

Para utilizar Google Maps dentro de Angular se utilizó la librería “Angular Google Maps”⁴⁹ (AGM), la cual contiene los componentes necesarios para poder integrar Google Maps. Luego se registró MusycGo como aplicación en “Google Maps Platform” para obtener un “API key” para finalmente poder usar los mapas de Google.

El mapa en su conjunto es la representación de una serie de eventos y funcionalidades que son gatilladas por la interacción del usuario, se detallan a continuación:

- El mapa ofrece una primera vista (nivel de *zoom* más alejado) que muestra las georreferenciaciones agrupadas por país (si un país no tiene canciones no es mostrado), la agrupación se realiza con una consulta a la base de datos para el atributo “country” en la colección de música. Al seleccionar un marcador se despliegan las canciones (sin repetición y que tengan el enlace de Spotify) ordenadas alfabéticamente por el nombre del artista (esto permite ver sucesivamente todas las canciones de un artista). Con el mismo marcador se puede acceder al perfil maestro de una canción.
- Cuando se modifica el estado inicial del *zoom* se pueden ver los marcadores de las georreferenciaciones. Si existen varios marcadores que se encuentran conglomerados en un sector específico del mapa, estos son agrupados (*clusters*) en marcadores (forma radial de color azul) que muestran el número de canciones que está agrupando. Cuando se selecciona un grupo, el *zoom* del mapa cambia para mostrarlos separadamente.
- Si existe más de un marcador posicionado en el mismo lugar o muy cercanos, estos son solapados, pero cuando se selecciona alguno de estos, se genera un evento que separa los marcadores en forma de “espiral”. Mientras más marcadores existan, el espiral se irá expandiendo para poder hacerlos todos visibles y seleccionables.
- El sistema está programado para mostrar solo los marcadores que están dentro del rectángulo del mapa que es visible por los usuarios (no aplica en los marcadores de países), por lo tanto, los marcadores son visibles cuando el usuario se desplace por el lugar. Para saber qué marcadores están dentro, se utiliza la función “google.maps.Polygon” y los cuatro vértices del mapa para generar un rectángulo virtual, y luego por cada marcador se aplica la función “containsLocation” que detecta si está dentro de las fronteras del rectángulo, si se cumple, se agrega a un arreglo de marcadores visibles.
- Cuando se realiza un evento de clic en cualquier lugar del mapa, se genera dinámicamente una lista con las cinco canciones cercanas al lugar seleccionado (se buscan canciones hasta tener cinco), y otra lista que muestra las canciones del país. Cabe señalar que las listas como tienen el fin de ser reproducidas solo son mostradas las que tienen el enlace de la canción en Spotify. Además, en cada elemento de las listas se puede acceder al perfil maestro de la canción.

⁴⁹ <https://angular-maps.com/>

- Para calcular distancias entre dos puntos, se utilizan fórmulas matemáticas que transforman las dos coordenadas en un resultado numérico que representa la distancia en metros.
- Cuando se posiciona el cursor sobre un marcador de una canción, se activa el evento que muestra tres ventanas de información, una que muestra el nombre del artista y de la canción, la otra ventana muestra la justificativa de la georreferenciación y una tercera muestra el nombre del lugar (en pequeño).

Otras funcionalidades del mapa se encuentran en sus botones superpuestos, se detallan a continuación:

- Al igual que Google Maps, se mantienen los botones para modificar el *zoom*, el botón para acceder al “*Street View*” con la visión en 360° de las calles y lugares del mundo, y finalmente se encuentra los botones para acceder a la vista satelital y a la superficie del mapa con relieve. Al usar “*Street View*” la plataforma permite ver en tercera dimensión donde está ubicado el marcador del lugar, por lo que se puede conocer realmente el lugar físico que inspiró la canción (siempre y cuando esté disponible por Google Maps).
- El botón para acceder a la ubicación actual permite posicionar al usuario en el centro del mapa, mostrando automáticamente la lista de las canciones cercanas en un radio determinado.
- Cuando se accede a la geolocalización se genera una lista dinámica de las canciones cercanas, posicionadas en un radio especificado por el usuario (de 0 a 50 kilómetros).
- Cuando se realiza un desplazamiento en el mapa, cambian las canciones visibles, con ello se genera una lista dinámica de las mismas (las canciones se repiten tantas veces como lugares existan de esa canción).
- Para visualizar las tendencias en visualizaciones y reproducciones, se crean dos listas que contienen las cinco canciones con los valores más altos dado el atributo “*look*” y “*hear*” de cada canción en la base de datos.
- En el botón de búsqueda del mapa existen dos cuadros de texto, el primero corresponde a la ubicación, que permite ingresar cualquier lugar del planeta, el sistema desplegará la lista de coincidencias y al seleccionar uno, el mapa es redirigido y centrado específicamente en ese lugar, lo cual permite ver las canciones cercanas. La funcionalidad para buscar la ubicación es parte de la API de Google Maps, a través de “*Geocoder*”, “*places.autocomplete*” y “*places.PlaceResult*”. Mientras que el segundo buscador contiene una tabla con todas las canciones de la base de datos y puede ser filtrada según el texto ingresado.
- En la esquina superior derecha del mapa existe un número que muestra la cantidad de marcadores visibles en el mapa.
- Al costado del número de canciones visibles hay un botón que actualiza el arreglo de marcadores en el mapa, que puede ser usado en el caso que no se estén visualizando correctamente.
- En la parte inferior del mapa se ve un botón que al presionarlo muestra la simbología de los marcadores y botones del mapa.

- Finalmente está el botón de configuración del mapa, el cual tiene dos usos, el primero es para cambiar el estilo y diseño del mapa, y el segundo uso es para filtrar los datos dentro del mapa. Existen tres seleccionadores múltiples para: artistas, género musical e idioma de la canción. Para cada filtro se encuentra definida una función para validar si cada elemento del arreglo de marcadores cumple con los filtros seleccionados, para cada elemento se ejecutan funciones anidadas, por lo que, si no cumple con alguna, quiere decir que el filtro para alguno de los atributos de la canción fue deshabilitado e inmediatamente el marcador es “rechazado” y luego no es mostrado en el mapa.

4.4.2. Estadística

Este componente reúne una serie de gráficos, listas y datos que muestran el uso que se le da a la plataforma (ver Figura 45).

Utilizando la librería “Chart.js”⁵⁰ y los datos recopilados de la base datos, se muestran las estadísticas de:

- **Los usuarios que han realizado más colaboraciones:** A través de un gráfico tipo *donuts* se muestran los usuarios, el número aportes define el área del gráfico que ocupa cada uno. Además, en un costado se muestra una lista descendiente tipo *ranking* de los usuarios (se muestra solo su *nickname*), siendo los tres primeros destacados con un estilo de letra y color de fondo. También existe un botón que permite acceder a la información básica de ese usuario y al listado de todas las colaboraciones que ha realizado. También se encuentra el conjunto de usuarios anónimos.
- **Número de canciones por país:** Con un gráfico de barras se muestran horizontalmente el número de canciones registradas dentro de los límites de cada país. En el costado derecho se muestra el listado descendente de los países (se incluyen también las banderas).
- **Las canciones que han sido más veces agregadas a favoritos:** Con un gráfico de barras se muestra un *ranking* de las canciones favoritas entre todos los usuarios. En el costado derecho se muestra un listado descendente indicando el nombre de la canción, artista y el número de veces que ha sido agregada a favoritos.
- **Las canciones que tienen dos o más lugares registrados:** Con un gráfico de barras descendente se muestran las canciones que tienen varios lugares registrados. También se muestra el listado en un costado.

4.4.3. Ingreso de canción

En esta sección de la aplicación se agregan las georreferenciaciones (ver Figura 43), un usuario puede ingresar una canción según dos modalidades (es opcional el inicio de sesión previo).

⁵⁰ <https://www.chartjs.org/>

El ingreso por un enlace de Spotify permite obtener la información básica de la canción (nombre, artista y género musical), mientras que los campos a ingresar son: lugar y justificación. Con el mapa proporcionado se selecciona uno o varios lugares, luego se selecciona la razón de la geolocalización (p. ej.: mención en el título, mención en la letra, referencia a un personaje local, etc.) y un campo de justificación (se puede incluir fragmentos de la letra, citas, enlaces, etc.). Opcionalmente se puede incluir la letra completa.

El ingreso por el formulario completo permite escribir todos los campos manualmente (nombre canción, artista, letra, género musical, enlace de YouTube, de Spotify y de SoundCloud). El mapa y la forma de ingresar el lugar es similar a la modalidad anterior.

Para los otros atributos de la canción que no son solicitados al usuario, la plataforma rellena por defecto algunos y otros son obtenidos por el componente (ejemplo: identificador del usuario, fecha y hora actual).

El sistema detecta si la canción ya ha sido registrada en un lugar ya georreferenciado y muestra un mensaje en pantalla con los lugares repetidos. Cuando un usuario ingresa una canción a través de un enlace de Spotify, el sistema valida si existe el enlace en la base de datos, si existe compara las coordenadas del nuevo lugar con las que existen en la base de datos previamente, si no se encuentran coincidencias se considera como un nuevo lugar. Mientras que cuando se ingresa por el formulario completo, el sistema valida si existe coincidencia en la base de datos con respecto a: nombre de la canción, artista, latitud y longitud (insensibilidad a mayúsculas y minúsculas en nombre de la canción y artista).

También se adiciona una tabla que permite agilizar el proceso de incluir un lugar a una canción ya registrada, por ejemplo, se puede seleccionar una canción y el sistema desplazará al usuario hacia el ingreso de la canción con el enlace de Spotify llenado por defecto.

4.4.4. Autenticación

Como se mencionó anteriormente, el inicio de sesión y registro de los usuarios es externalizado al servicio ofrecido por Auth0 (ver Figura 33).

En MusycGo existe el botón de inicio de sesión, al momento de ser presionado la página actual es redirigida al servicio de Auth0, luego se ofrece la autenticación a través de Facebook, Gmail, Twitter o por el clásico modo de correo y contraseña. Cuando el usuario selecciona o introduce sus datos, el servicio redirige al usuario a MusycGo, el cual toma los datos de la URL (*token* de acceso, tiempo de expiración y la información del usuario), que luego de ser interpretada por el sistema, reconoce al usuario. Si el usuario ya existe en la base de datos de MusycGo se toman sus datos, caso contrario, si el usuario no existe, es agregado a la base de datos.

Cuando el sistema identifica al usuario, verifica si es o no un moderador y establece los privilegios según el caso.

4.4.5. Notificación

Las notificaciones corresponden al historial de las canciones registradas por un usuario que han sido modificadas. El historial se muestra en orden descendente con respecto a la fecha y hora de la modificación (ver Figura 58).

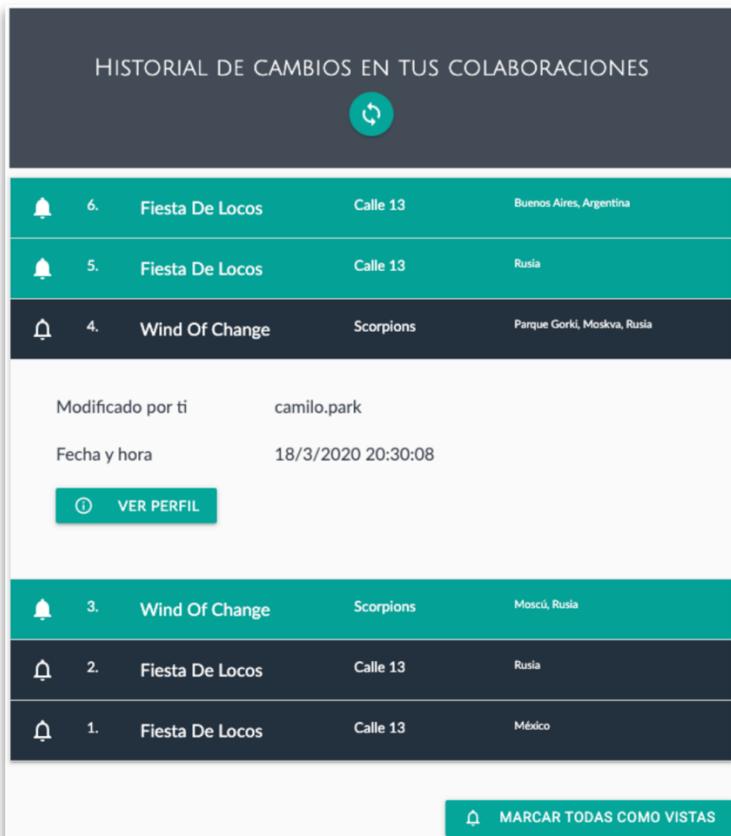


Figura 58. Notificaciones de cambios en colaboraciones

En el momento que un usuario autenticado realiza un cambio desde el “perfil del lugar de la canción”, este cambio es notificado al usuario que registró dicho lugar. Una canción puede tener lugares registrados por diferentes usuarios, por esto la notificación de un cambio llega solo al usuario que le corresponda su lugar registrado y no de lugares registrados por otros.

Una notificación contiene: *nickname* del usuario que la modificó, nombre de la canción, artista, lugar, fecha y hora, y un botón que redirige al “perfil del lugar de la canción”.

Las notificaciones permanecen como “no vistas” hasta que el usuario las marque como “vistas”, a través de un botón o simplemente cuando se seleccione cada notificación.

4.4.6. Buscador

En la plataforma existen dos lugares donde se accede a un buscador, desde el mapa y desde el menú general de la pantalla principal de la aplicación (buscador general, ver Figura 44). En ambos casos los datos mostrados son las georreferenciaciones en un estado “activo”. También se puede: ordenar las filas según el criterio de alguna de sus columnas, ver en el pie de la tabla el número total de registros, además existe un campo donde se puede ingresar una cadena de texto que filtra los resultados según la coincidencia en alguna de las columnas de cada registro.

El buscador general tiene la particularidad de que muestra los campos que el usuario omitió al momento de registrar un lugar de una canción, esto puede incentivar a los usuarios a complementar la georreferenciación. También existe el botón para acceder rápidamente al “perfil del lugar de la canción”.

Las tablas están confeccionadas gracias al *plugin* para JQuery llamado “DataTable”⁵¹, el cual permite añadir funcionalidades avanzadas a las tablas HTML.

4.4.7. Sitio 404

Esta página de estilo minimalista es mostrada al usuario cuando accede a una URL inválida o no encontrada según las rutas definidas en Angular. La ventana muestra un botón “volver” que permite redirigir a la pantalla principal de MusycGo (ver Figura 59).



Figura 59. Ventana de URL inválida

4.4.8. Perfil de usuario

Existe el perfil de usuario que solo es visible para el usuario actualmente autenticado (ver Figura 49) y otro que es visible por la comunidad (ver Figura 60). Para el caso del perfil

⁵¹ <https://datatables.net/>

privado se muestra la información básica del usuario, las colaboraciones realizadas y las canciones guardadas en favoritos, mientras que para el perfil público se muestra: imagen del usuario, *nickname*, botones para compartir el perfil y las colaboraciones realizadas.

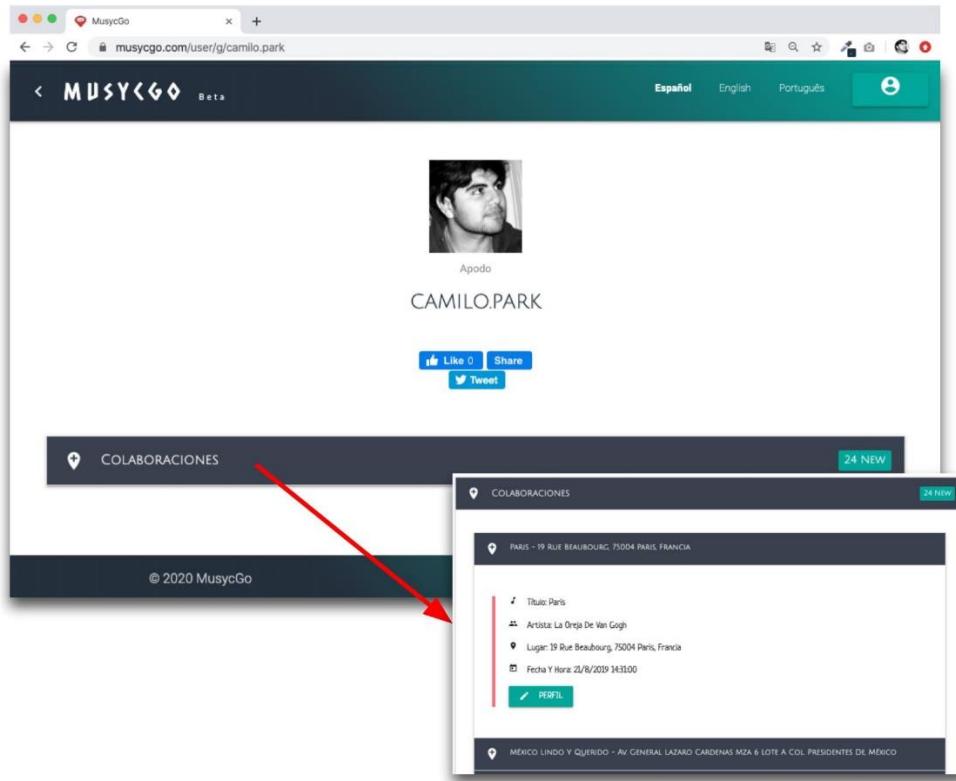


Figura 60. Perfil público de usuario

Para las canciones visibles en estos perfiles se puede ver la información básica y también se puede acceder al “perfil del lugar de la canción”.

El perfil privado es el único lugar donde un usuario puede “eliminar” sus colaboraciones.

La URL del perfil privado es “<https://www.musycgo.com/user>” (usuario previamente autenticado en la plataforma), mientras que la URL del perfil público está constituido por: “<https://www.musycgo.com>” + “/user/” + “[‘g’, ‘f’, ‘t’ o ‘a’]” + “[nickname]”.

4.4.9. Perfil maestro de la canción

Este componente agrupa en un perfil la información de una canción en particular (ver Figura 53). Se puede ver: imagen de la canción y del artista, nombre de la canción, nombre del artista, botones de compartir en redes sociales, enlaces a contenido multimedia y para acceder a la letra en “Musixmatch”. Se proporciona la letra ingresada por el usuario. También se muestra un mapa con todos los lugares donde se encuentra georreferenciada

la canción y se adjunta: la lista con los lugares, la justificación, el *nickname* del usuario que registró la justificación y un botón para acceder al perfil del lugar.

Dentro de este perfil se pueden modificar ciertos atributos que son parte de la canción, tales como: enlace de YouTube y SoundCloud, la letra y el idioma de la canción.

La URL del perfil maestro se compone de: “<https://www.musycgo.com>” + “/master/” + “[identificador de alguno de los lugares de la canción]”, es decir, es posible acceder al mismo perfil maestro dado cualquier identificador del lugar de la canción.

Los lugares que se ven en este perfil son obtenidos según una consulta a la base de datos en la colección de música, la cual realiza la agrupación de una canción si coinciden en tener el mismo nombre de la canción y del artista. Para saber que nombre de artista y que nombre de canción buscar, el sistema toma el identificador en la URL y posteriormente realiza la búsqueda en la base de datos en la colección de música dado ese identificador, luego se obtiene el nombre de la canción y del artista.

4.4.10. Perfil del lugar de la canción

En este perfil se puede obtener información de la georreferenciación, modificar el lugar de la canción, compartir en las redes sociales, reproducir, calificar y comentar (ver Figura 50, Figura 51 y Figura 52).

En primer lugar, este perfil accede a la API de Spotify para obtener la imagen destacada de la canción. Luego asigna los datos que provienen de la base de datos con respecto a la canción actual (número de veces que ha sido agregada a favoritos, número de visualizaciones, número de reproducciones, lugar, letra, justificación y géneros musicales). También asigna los enlaces a los diferentes reproductores multimedia, en caso de que no se provea el enlace, el sistema define una URL compuesta del nombre de la canción y del artista, que permite la búsqueda en las plataformas externas (Spotify, YouTube o SoundCloud) y también para conocer la letra en el sitio web de “Musixmatch”.

Como información extra, se incluye un mapa que muestra los demás lugares donde se encuentra la canción y al costado se puede ver un listado de estos lugares (se incluye botón para acceder al perfil del lugar).

En este perfil se puede conocer el historial de justificaciones que ha tenido una canción. Dado que el modelo de datos define la justificación como un conjunto de justificaciones (justificación, usuario, fecha y hora), se pueden desplegar en una ventana emergente. Si solo existe la justificación actual se muestra un mensaje que indica que no se han encontrado versiones anteriores.

Otra funcionalidad es la modificación de la información del lugar o de la canción, a través del formulario que permite cambiar los datos de lugar, razón de la geolocalización, justificación, letra, enlace a YouTube, SoundCloud e idiomas a la canción. Este formulario se encuentra por defecto llenado con los datos actuales de la canción, pero mientras no

se modifique dicha información por defecto, no se podrán guardar los datos. Por otra parte, cabe señalar que los campos: lugar, razón y justificación, son solo modificados en el perfil actual del lugar de la canción (atributos del lugar), mientras que si se modifica: la letra, enlace a YouTube, SoundCloud o el idioma de la canción, son modificados en todos los lugares donde se encuentra la canción (son atributos de la canción).

Al pie del perfil se encuentra la sección “¿Qué Opinas?”, la cual permite calificar y comentar la georreferenciación. Las calificaciones se utilizan con el fin de saber cómo percibe la gente la pertenencia de la canción en el lugar, con la pregunta “Califica la pertenencia de la canción en este lugar”, cuyas posibles respuestas son: “Absolutamente”, “Probablemente”, “Medianamente posible”, “Poco probable”, “No pertenece” o “No lo sé”. Hay que señalar que el usuario debe estar autenticado para que pueda seleccionar una opción única, además se adjuntan las respuestas totales con respecto a ese lugar. Para la sección de comentarios y previa autenticación del usuario en la plataforma “Disqus”, se pueden generar hilos de comentarios e interacción entre los usuarios, cuyos comentarios estarán ligados al perfil actual del lugar de la canción.

Hay que señalar que en este perfil un usuario de tipo moderador puede desvincular cualquier canción de la plataforma, a través del botón “Eliminar”.

La URL del “perfil del lugar de la canción” se compone de: “<https://www.musycgo.com>” + “/place/” + “[identificador del lugar de la canción]”.

4.5. Otros recursos

Dentro del desarrollo existen módulos, herramientas, bases de datos y servicios que ayudaron a cumplir una gran variedad de funcionalidades y características fundamentales de MusycGo, estos se detallan en las siguientes subsecciones.

4.5.1. Multi-idioma

Para incluir varios idiomas en la gran mayoría de los textos que muestran la aplicación (otras traducciones las ofrecen automáticamente DataTable, Disqus y Google Maps), se instaló la librería de internacionalización para Angular llamada “ngx-translate”⁵², la cual ofrece un servicio, una directiva y un pipe para manejar el contenido de texto en la plataforma.

Para cada idioma se define un archivo JSON en la carpeta “i18n” dentro del proyecto. Dado que el sistema consta de tres idiomas (español, inglés y portugués), se tienen tres archivos JSON separados. En dichos archivos se encuentra una estructura “clave: valor”, en donde la “clave” es el texto (exacto) que compara “ngx-translate” con el texto definido en MusycGo, si coinciden se muestra en pantalla el texto que está definido en el “valor”. Para que el servicio de traducción sepa cuál archivo de idioma buscar, los componentes

⁵² <http://www.ngx-translate.com/>

capturan la variable del idioma actual (por defecto en español) o cuando es cambiado por el usuario en la cabecera de las ventanas de MusycGo.

A los textos en el código HTML se le asignó un pipe llamado “translate”, por lo que la estructura de un texto en el código tiene la siguiente forma: “{{ “Texto a cambiar de idioma” | translate }}”, lo que permite traducir específicamente los textos seleccionados.

Cabe señalar que la estructura “clave: valor” para cada texto fue definido por el tesisista en conjunto con el patrocinante. Las traducciones están en constante refinamiento, incluso los usuarios pueden entregar posibles mejoras a través del formulario de contacto dentro del sistema.

4.5.2. Estilos de mapa

MusycGo a primera vista se muestra como un mapa de gran tamaño, constituido por un estilo y colores concordantes con el resto de la plataforma, además de otros estilos de mapas, los cuales se pueden modificar desde el botón de configuración del mapa (ver Figura 42). Para lograr estas paletas de colores se utilizó una plataforma web llamada “Snazzy Maps”⁵³, la cual tiene un amplio catálogo de estilos tanto de pago, como gratuitos, incluso cuenta con un “constructor” donde el usuario (previamente autenticado) puede crear sus propios diseños, y fue esta modalidad la utilizada, donde se diseñaron variados estilos que fueron exportados en formato JSON para luego ser añadidos a un archivo TypeScript como constantes del sistema para su posterior uso.

4.5.3. Base de datos de países

La base de datos de países surge por la necesidad de agrupar las canciones por países y mostrarlos en el mapa, traducir los nombres según el idioma seleccionado y también para añadir las banderas en casos particulares.

Luego de un periodo de revisión y búsqueda de posibilidades dentro de Google Maps, no se logró corroborar que se pudieran cumplir todas las necesidades, en ese momento surge una alternativa muy completa y que cumplía las necesidades. A través del sitio web llamado “Rest Countries”⁵⁴ se pudo acceder a una API de los datos de todos los países del mundo. Con la URL “<https://restcountries.eu/rest/v2/all>” se accedió a todo el contenido de los países en formato JSON, cuyos datos fueron tomados y agregados a un archivo TypeScript para luego ser accedidos dentro del proyecto sin la necesidad de realizar el llamado a la API.

Cuando se agrega una nueva georreferenciación en la plataforma, se captura con Google Maps el nombre del país de forma abreviada (independiente del idioma siempre es igual), luego se busca dentro de la base de datos de países las coordenadas geográficas del país para luego ubicar un marcador sobre dicho país en el mapa. Para obtener el nombre

⁵³ <https://snazzymaps.com/>

⁵⁴ <https://restcountries.eu/>

completo del país se utiliza un pipe llamado “fullnamecountry”, el cual busca la “clave” llamada “alpha2code” en la base de datos de países y si es igual al nombre abreviado retorna el “valor” correspondiente al nombre del país en el idioma seleccionado. Caso similar ocurre con la obtención de la URL de la imagen de la bandera, ya que se ejecuta otro pipe llamado “getflag” que busca la misma “clave” llamada “alpha2code” y retorna la URL.

4.5.4. Servicio de Spotify

El uso de Spotify dentro de MusycGo surge para ofrecer datos, información y música reproducible a los usuarios que visitan e interactúan con el sistema.

Basado en consultas a la “Web API de Spotify”⁵⁵, se obtiene como respuesta metadatos en formato JSON sobre los artistas musicales, álbumes y canciones, directamente desde el catálogo de datos de Spotify.

Para poder acceder a la API de Spotify se debió autorizar la aplicación, a través del “Flujo de Credenciales del Cliente” (*Client Credentials Flow*). Este flujo se usa en la autenticación de servidor a servidor y permite acceder a los “puntos finales” de la API (métodos HTTP que especifican dónde las API pueden acceder a los recursos), pero con la restricción de que no se puede acceder o administrar los datos de un usuario (Spotify for Developers, 2020). Para configurar este flujo de autorización se inició sesión en “Spotify for Developers” y luego se registró la aplicación, se obtuvo un “Client ID” y “Secret Key” que fueron agregados como constantes en el servidor, lo que permite que cuando el servidor web inicie, genere un “Access token” automáticamente, lo que a fin de cuentas conlleva a evitar que el usuario deba realizar un inicio de sesión en Spotify antes de usar MusycGo.

Los “puntos finales” de la API que fueron mayormente utilizados son:

- A. [https://api.spotify.com/v1/tracks/\\${id_track}](https://api.spotify.com/v1/tracks/${id_track})
- B. [https://api.spotify.com/v1/artists/\\${id_artist}](https://api.spotify.com/v1/artists/${id_artist})

Siendo “id_track” el identificador proporcionado en el enlace de una canción de Spotify, por ejemplo, en el enlace: “<https://open.spotify.com/track/3D4iziSjHi4jxG8r4pta7z>”, el fragmento de texto destacado corresponde al identificador. Por otro lado, “id_artist” es obtenido dentro de la respuesta del “punto final” A (identificador del artista que creó la canción consultada).

El “punto final” A permite recuperar todos los datos de la canción con identificador “id_track”. Mientras que el “punto final” B retorna los datos de un artista con el identificador “id_artist”.

⁵⁵ <https://developer.spotify.com/>

Para ambos “puntos finales” que son ejecutados desde Angular (servicio “spotify.service”) se incluye el “Access token” generado en el servidor y luego las respuestas hacia Angular son en formato JSON.

Otra utilidad de Spotify dentro de MusycGo es poder reproducir una canción, el proceso para poder escucharla es:

1. Incluir el enlace de la canción en Spotify en la georreferenciación.
2. En el código HTML se incluye la etiqueta “<iframe>”, cuyo atributo “src” debe incluir el enlace de la canción ligeramente modificado (paso 3).
3. A través del pipe “linkconverted” agrega el fragmento “embed” dentro de la URL. Por ejemplo:
 - a. **Antes:** <https://open.spotify.com/track/3D4iziSjHi4jxG8r4pta7z>
 - b. **Después:** <https://open.spotify.com/embed/track/3D4iziSjHi4jxG8r4pta7z>
4. Según la preferencia se pueden cambiar otros atributos de la etiqueta “<iframe>”, por ejemplo, los atributos “width” y “height” (ancho y alto del reproductor).
5. Finalmente se genera un reproductor como se puede observar en la Figura 27.

4.5.5. Geolocalización

La geolocalización está disponible para todos los usuarios, en especial para aquellos usuarios que quieran conocer las canciones cercanas a su ubicación (p. ej., viajeros amantes de la música). Mientras se desplazan pueden ir actualizando su posición para descubrir nuevas canciones a su alrededor, las cuales se agrupan automáticamente en la lista “Mis Canciones Cercanas”.

Para capturar la geolocalización del usuario, se hace uso de la API de Geolocalización⁵⁶, que provee información sobre la ubicación física del dispositivo.

Cuando se accede a la ubicación desde el botón proporcionado en el mapa, el navegador web solicitará al usuario que permita el acceso a la ubicación la primera vez que se requiera su uso.

Para saber si el navegador soporta la API, se creó un condicional que retorna una alerta que indica si el navegador la soporta o no. En caso de ser válido, se ejecuta el método “getCurrentPosition()”, el cual inicia una solicitud asíncrona para detectar la posición actual del usuario y devuelve un objeto con las coordenadas de latitud y longitud.

Para trabajar con esta API se configuró un protocolo criptográfico para comunicaciones seguras en el dominio de MusycGo (certificado SSL).

⁵⁶ https://developer.mozilla.org/en-US/docs/Web/API/Geolocation_API

4.6. Revisión de la versión lograda

Considerando todas las historias de usuarios, funcionalidades y características desarrolladas, el *Scrum Master* determinó que la versión lograda cumple con ser un prototipo funcional para el proyecto de título y también para ser difundido.

Dentro del desarrollo del proyecto se revisó la ejecución de las 11 prácticas de Integración Continua (IC), las cuales permitieron: reducir los riesgos, conocer en todo momento el estado del proyecto y eliminar las barreras de desplegar frecuentemente. A continuación, se detalla cómo se abordó cada una.

1. **Mantener un único repositorio fuente:** Solo existe un repositorio privado alojado en Bitbucket, en el cual se versionó todo el proyecto.
2. **Automatizar el *build* (compilación):** Una tarea automatizada en Jenkins es el encargado de compilar el proyecto para cada cambio en el repositorio.
3. **Hacer el *build* auto-testeable:** Aplicación parcial de análisis estático mediante la identificación y resolución automática de patrones problemáticos encontrados en el código fuente, aplicando reglas personalizadas vía ESLint⁵⁷.
4. **Todos realizan *commits* todos los días:** Al ser solo un desarrollador, el *commits* diario fue realizado por varios meses, mas no toda la duración del proyecto. Esta práctica fue impulsada para satisfacer el entendimiento de las reuniones diarias de la metodología Scrum.
5. **Cada *commit* gatilla el *build* en una máquina de integración:** Como se mencionó anteriormente, existe una tarea en Jenkins que realiza esta práctica.
6. **Corregir *builds* fallidos inmediatamente:** Si los *builds* fallaban en el proceso eran corregidos antes de ser desplegados en producción.
7. **Mantener el *build* rápido:** Tomando cinco *builds* (entre el 4 de febrero y el 9 de febrero del 2020) ejecutados según la tarea definida en Jenkins para dicho proceso, el tiempo promedio que tardaron fue de 1 minuto y 49 segundos. Según (Fowler, 2006) “una construcción de diez minutos está perfectamente dentro de lo razonable”.
8. **Ejecutar pruebas en una máquina clon a la del entorno de producción:** Dado que no existía una máquina clon que tuviera las mismas configuraciones, versiones y sistema operativo, no se abordó esta práctica.
9. **Todos pueden obtener la última versión estable de la aplicación:** En el momento que se despliega la aplicación en producción, cualquiera con conexión a Internet y que tenga el enlace de MusycGo puede acceder a la última versión de la aplicación. Igualmente, el repositorio (rama *master*) ofrece la versión estable de la aplicación.
10. **Todos pueden ver qué está pasando:** Para esta práctica existieron varias formas de ver el estado, estas son: dado los comentarios enlazados al versionamiento del código en el repositorio, también usando la aplicación de Trello donde se puede ver las funcionalidades en proceso y las que están terminadas, y finalmente a través

⁵⁷ <https://eslint.org/>

de Jenkins, en donde existen *flags* que indican el estado de las compilaciones (exitosas y fallidas).

11. **Automatizar el *deployment*:** El despliegue está automatizado a través del servidor de Integración Continua (Jenkins).

De forma complementaria se creó un manual que describe los pasos, tareas y configuraciones sobre el proceso de IC llevado a cabo, el cual se encuentra en posesión del equipo del proyecto.

5. VALIDACIÓN Y RESULTADOS

5.1. Nombre de la plataforma

Para la selección del nombre de la plataforma se realizó el siguiente proceso:

- Generación de lista de nombres tentativos a través de un *brainstorming* inicial.
- Especificación de aspectos relevantes para evaluar la pertinencia de estos nombres: "¿Memorable?", "¿Breve?", "¿Diferenciador?", etc.
- Evaluación de distintas personas sobre los nombres tentativos y otros de su preferencia.
- Los que obtuvieron la mayor puntuación fueron: "MusicGo" y "GeoMusic", de los cuales fue seleccionado "MusicGo".
- Luego al encontrarse con coincidencias de nombres de usuarios en redes sociales registrados con dicho nombre se decidió buscar variaciones disponibles del mismo, hasta llegar a "MusycGo".
- El nombre "MusycGo" nace por el hecho de tener un nombre fácil de recordar y que sea descriptivo en su funcionalidad. "Musyc" es una adaptación de "Music" ("Música" en español), mientras que "Go" ("Ir" en español) hace alusión a descubrir, a ir tras la música, conocer lo que te rodea tanto físicamente como digitalmente a través de la plataforma MusycGo.

5.2. Test de usabilidad

Usabilidad se refiere a la calidad de la experiencia del usuario cuando se produce una interacción entre el usuario y el producto o sistema. La usabilidad engloba tres conceptos primordiales: efectividad, eficiencia y satisfacción del usuario.

Las pruebas de usabilidad se centran en que tan bien un usuario puede aprender y usar un producto para lograr ciertos objetivos.

La manera de recopilar la información varía según el método de evaluación escogido, para el caso particular de este proyecto de título se optó por utilizar la herramienta de "Escala de Usabilidad del Sistema" (*System Usability Scale*, SUS).

SUS, creado por John Brooke en el año 1986, es considerada una herramienta rápida y confiable para medir la usabilidad. Consiste en un cuestionario de diez ítems con cinco opciones de respuesta (de "Totalmente en desacuerdo" a "Totalmente de acuerdo"), con la que se puede evaluar una gran variedad de productos y servicios (Usability.gov, 2003).

El SUS ha sido probado por más de 30 años, y ha demostrado ser un método confiable para evaluar la usabilidad de los sistemas en comparación con los estándares de la industria.

5.2.1. Elección del grupo

Esta prueba de usabilidad se aplicó de manera remota, es decir, se seleccionó un grupo de personas conocidas y se les consultó de manera digital (correo electrónico y redes sociales) sobre la posibilidad de efectuar con ellos la prueba de usabilidad de MusycGo. Para las personas que aceptaron se les hizo entrega de los enlaces de la aplicación y del cuestionario confeccionado en Google Forms (ver Anexo A). A estas personas se les solicitó probar libremente el sistema por cinco días y luego debían contestar el cuestionario.

El grupo estuvo conformado por siete personas que ayudaron en primera instancia en el acercamiento al nombre final de la aplicación, por lo que, a modo de agradecimiento por su colaboración, ellos fueron los primeros en ser consultados para realizar esta prueba, además ellos tuvieron la posibilidad de ser los primeros usuarios en probar MusycGo como un prototipo en fase *beta*. Por otro lado, tenemos a 14 personas más que fueron consultados, ocho de ellos eran estudiantes y recién egresados de carreras de Informática, otras tres personas eran externas al área Informática y finalmente tres personas extranjeras.

La elección de 21 personas para que utilicen el sistema por cinco días permitió limitar la diseminación y distribución no autorizada (finalmente fueron 16 personas que utilizaron el sistema).

5.2.2. Desarrollo

Como se mencionó anteriormente la aplicación se encontraba en fase *beta* al momento de realizar la prueba de usabilidad, es decir, si bien las funcionalidades principales y con alto grado de prioridad ya se encontraban implementadas, era la primera vez que la aplicación era lanzada fuera del equipo del proyecto, por lo que podría haber tenido errores y problemas técnicos, es decir, no estaba garantizado el funcionamiento en su totalidad.

Cuando se les entregó los enlaces de la aplicación y del cuestionario se les explicó el contexto del proyecto brevemente. Luego se les solicitó que probaran el sistema el tiempo que necesiten y las veces que quieran por cinco días, recordando contestar el cuestionario entregado.

Además de los diez ítems del SUS, al cuestionario se agregaron nueve preguntas destinadas a caracterizar los usuarios y saber un poco más de sus interacciones con el sistema. Debido a que era una prueba de usabilidad ejecutada de manera remota y las respuestas del cuestionario eran anónimas, fue necesario formular preguntas extras para obtener más información.

Las preguntas fueron reformuladas incorporando el nombre de la aplicación en base al SUS (Measuring The User Experience) y se muestran a continuación:

1. Creo que me gustaría usar MusycGo con frecuencia.
2. Encontré MusycGo innecesariamente complejo.

3. Pensé que MusycGo era fácil de usar.
4. Creo que necesitaría asistencia para poder usar MusycGo.
5. Encontré que las diversas funciones en MusycGo estaban bien integradas.
6. Pensé que había demasiada inconsistencia en MusycGo.
7. Me imagino que la mayoría de la gente aprendería a usar MusycGo muy rápidamente.
8. Me pareció muy engoroso/incómodo de usar.
9. Me sentí muy seguro usando MusycGo.
10. Necesitaba aprender muchas cosas antes de poder usar MusycGo.

El formato de respuesta para los diez ítems del SUS se aprecia en la Figura 61.

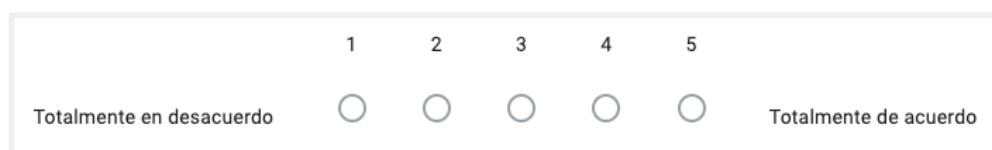


Figura 61. Formato respuesta SUS

Mientras que los ítems extras fueron:

1. Ingrese su edad.
2. ¿Cuánto tiempo usó MusycGo? (tiempo estimado en minutos).
3. ¿Cuál crees que sería tu perfil de usuario más representativo dentro de la aplicación? (“Usuario Explorador (Ver, escuchar y aprender sobre los lugares que inspiraron canciones)”, “Usuario Colaborador (Ver, escuchar, modificar y aportar con nuevos lugares que inspiraron canciones)” o “No usaría la aplicación”).
4. ¿Cuál crees que sería la mejor manera de promocionar y/o divulgar la aplicación?
5. ¿Alguna funcionalidad no se comportó como esperaba?
6. ¿Qué es lo que más te gustó y lo que menos te gustó de MusycGo?
7. ¿En qué navegador ingresó a MusycGo? (Chrome, Firefox, Safari, Opera, Otro).
8. ¿En qué dispositivo ingresó a MusycGo? (Móvil, Computador, Tablet, Otro).
9. Proporcione cualquier comentario extra sobre MusycGo.

Para poder interpretar los resultados del SUS se debe realizar el siguiente proceso:

1. Listar los números correspondientes a las respuestas para cada pregunta (para cada usuario participante).
2. A las respuestas de las preguntas impares se le resta 1 al número de la respuesta (Ejemplo: [resultado pregunta 1 del usuario 1] - 1).
3. Para cada una de las respuestas de las preguntas pares, reste su valor de 5 (Ejemplo: 5 - [resultado pregunta 2 del usuario 1]).
4. Sumar todas las respuestas de los pasos 2 y 3 por cada usuario.
5. Multiplicar el resultado del paso 4 por 2.5.
6. Finalmente, sumar los resultados del paso 5 y dividirlo entre el número de usuarios que respondieron el cuestionario.

7. El número obtenido se encuentra en el rango de 0-100, este puntaje SUS no se debe que confundir con porcentajes.

Un puntaje SUS superior a 68 se considera sobre promedio y cualquier puntaje inferior a 68 está por debajo del promedio, sin embargo, la mejor manera de interpretar los resultados consiste en "normalizar" los puntajes para obtener una clasificación percentil (Usability.gov, 2013).

5.2.3. Resultados

Los resultados se describirán en dos partes, la primera es para las preguntas de caracterización de usuarios y la otra parte para los resultados del SUS.

Dentro de los resultados para las preguntas extras, el histograma de edad de los participantes (ver Figura 62) muestra que el rango etario predominante estuvo entre 23-27 años, con seis usuarios. También cabe mencionar que la edad mínima fue de 18 años y la edad máxima fue 34 años.

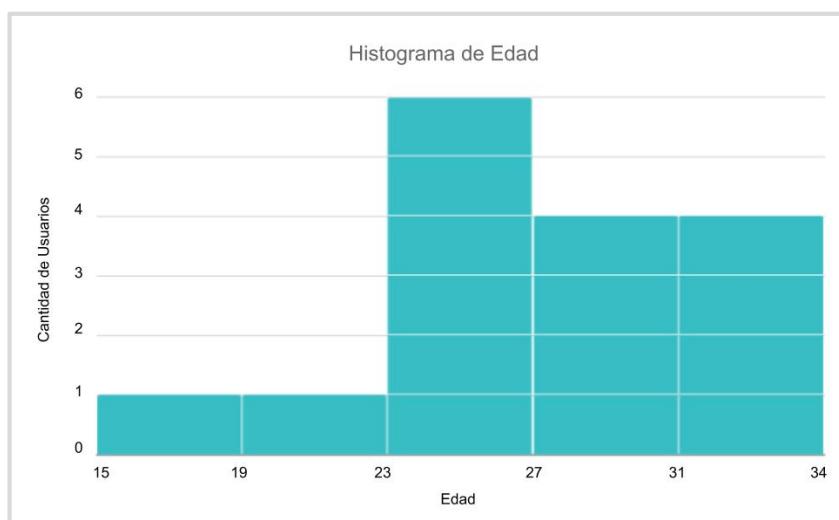


Figura 62. Histograma cantidad de usuarios en un rango etario

Sobre el tiempo que los participantes usaron la aplicación (ver Figura 63), existe un resultado interesante, ya que una persona dice haber utilizado MusycGo por dos horas, lo que indica un interés por la idea del proyecto y también por querer conocer todo sobre el funcionamiento. Mientras que 13 personas declaran haber usado el sistema por menos de 50 minutos. Como información adicional, los extremos de este resultado lo conforman: cinco participantes utilizaron el sistema por cinco minutos y un participante usó el sistema por 120 minutos.

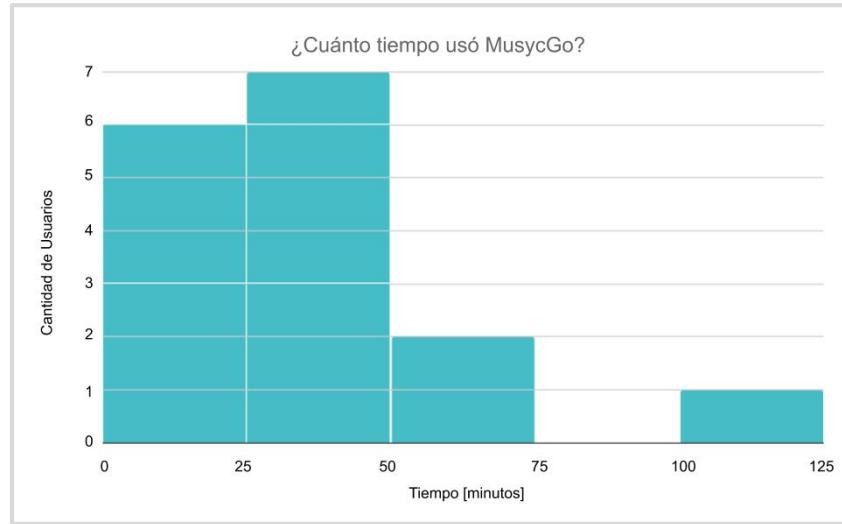


Figura 63. Cantidad de usuarios que usaron MusycGo por un rango de tiempo

Una información muy relevante fue conocer cuál era el perfil donde los usuarios se sentían más atraídos o el que los representa más (ver Figura 64). Sorprendentemente el resultado muestra que existe la misma proporción para ambos perfiles, y que además ninguna persona indicó que no usaría la aplicación.

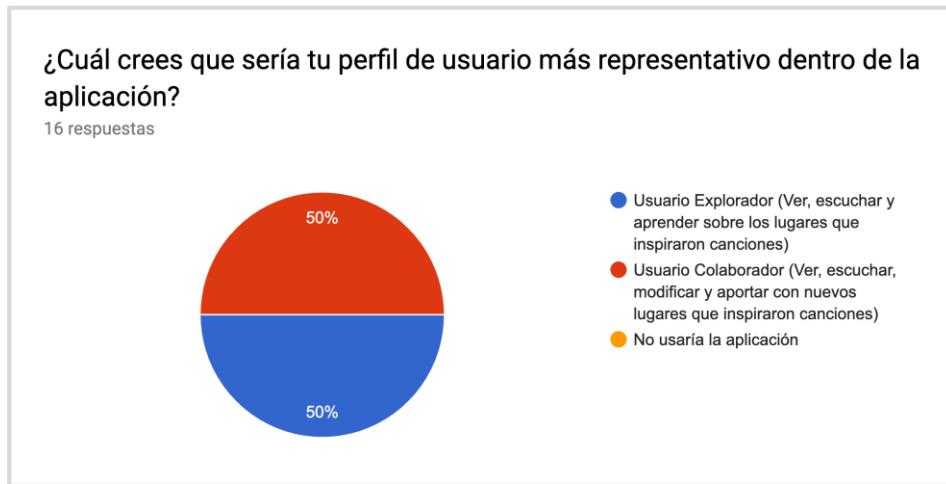


Figura 64. Porcentaje de usuarios según perfil

En cuanto al navegador utilizado para ingresar a la aplicación (ver Figura 65), el 75% de los participantes (12 personas) hicieron uso de Google Chrome, mientras que el 25% restante usaron otras alternativas.

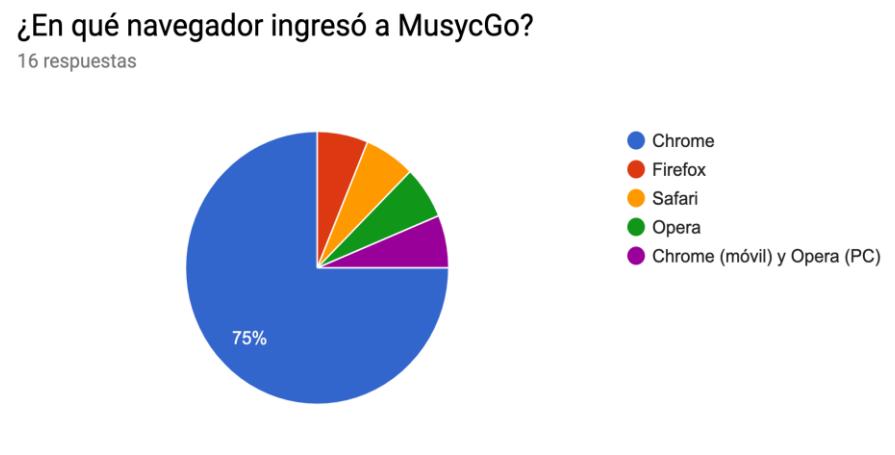


Figura 65. Porcentaje de ingresos a MusycGo por navegador

Con respecto al dispositivo que fue más utilizado para ingresar a MusycGo (ver Figura 66), el 62,5% de los participantes (diez personas) utilizaron un computador, la segunda posición indica que el 31,3% (cinco personas) ingresaron desde un dispositivo móvil y una persona usó ambas. Ninguna persona usó un *tablet* para acceder.

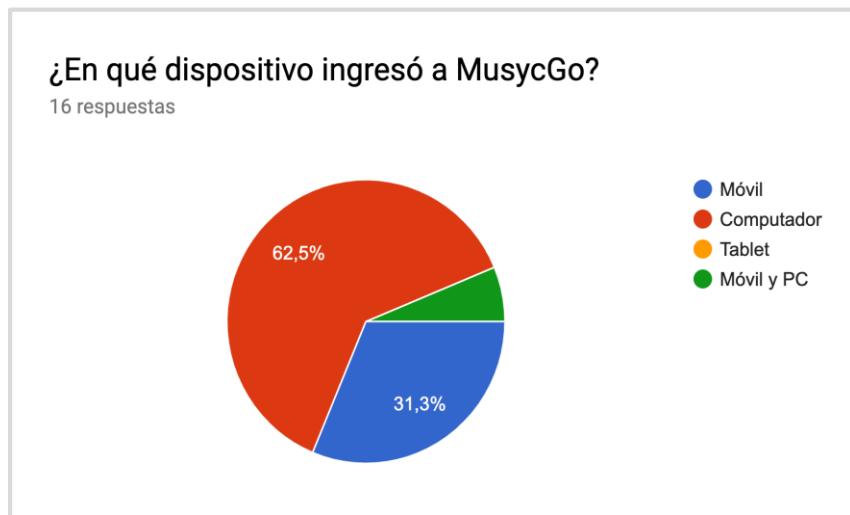


Figura 66. Porcentaje de ingresos a MusycGo por dispositivo

Para la pregunta que hace alusión a la mejor manera de promocionar y/o divulgar la aplicación, la respuesta común en la gran mayoría de las personas fue que a través de redes sociales. Otras personas señalan que incluir a los propios artistas en este proceso puede ser muy beneficioso, en donde ellos puedan ingresar sus canciones y luego dar a conocer a sus seguidores sobre su aparición en la aplicación. En ciertas respuestas se mencionan municipalidades, radios locales y encargados de turismo.

Sobre la pregunta que mencionaba si una funcionalidad no se comportó como esperaba, la mitad de las personas que respondieron mencionaron que todo estaba normal, otras comentaron algunos problemas, estos fueron: al momento de iniciar sesión tuvo que actualizar la página para que se reconociera su cuenta en la aplicación, en la versión móvil los botones laterales del mapa no reaccionan al tocarlos (menciona que quizás era por el tamaño de los botones), una persona esperaba que se pudiera registrar una canción en base al enlace de YouTube (funcionalidad no disponible por el momento), finalmente un usuario consideró que el mapa no carga rápidamente.

Lo que los usuarios respondieron a lo que más le gustó fue:

- Alto grado de interés por la idea de conocer y compartir canciones con relación a partes de todo el mundo.
- Una interfaz intuitiva, amigable y fácil de entender, con una paleta de colores visualmente agradables.
- Conocer un poco más de sus canciones favoritas.
- Poder saber a través de la agrupación de canciones, los lugares que más canciones tienen.
- Poder escuchar canciones mientras se navega por la aplicación.
- Poder personalizar el mapa.

Lo que menos les gustó a los usuarios fue:

- La obligación de escribir una justificativa para cada lugar al que se ingresa una canción.
- Los reproductores de la vista previa de la canción tardaban un poco en aparecer cuando había una lista de ellos.
- La animación interminable de algunos botones.
- No poder ingresar una canción directamente desde el mapa principal.
- En la versión móvil, para poder desplazar el mapa se tenía que hacer con dos dedos sobre la pantalla.
- Las canciones se sobreponen si se seleccionan más de una al mismo tiempo.
- En algunos casos existen muchos colores en el mapa.
- No se entendió del todo el funcionamiento del mapa y la utilidad de haber iniciado sesión.
- La actualización del mapa cada vez que se cambia una opción de los filtros del mapa.

Algunas ideas que surgieron de los usuarios fue que sería interesante incluir canciones típicas de los países y/o ciudades. También comentaban que para los formularios y la seguridad de la plataforma el uso de *captchas* para *bots* puede ser una idea que considerar. Por otra parte, señalan que sería bueno incluir formas de recompensas para motivar a usar constantemente la aplicación.

Los usuarios que probaron la aplicación mencionaron que la plataforma era de su agrado, con un gran potencial, indicando que es muy probable que la sigan utilizando.

Se mostrarán los resultados obtenidos del cuestionario SUS.

Las puntuaciones para cada pregunta se pueden ver en el Anexo B, mientras que las puntuaciones finales se muestran en la Tabla 20. La puntuación obtenida en la primera prueba de usabilidad según la herramienta SUS, fue de 70,3. Como se mencionó anteriormente en la sección 5.2.2, un puntaje de SUS superior a 68 se considera mejor que el promedio.

Tabla 20. Puntaje y puntuación SUS

Usuario	Suma Puntajes	Puntuación SUS
1	30	75
2	33	82,5
3	28	70
4	26	65
5	31	77,5
6	30	75
7	35	87,5
8	30	75
9	25	62,5
10	29	72,5
11	30	75
12	28	70
13	28	70
14	26	65
15	22	55
16	19	47,5
Total	450	70,3125

Según (Linh Giang, 2019), se puede calificar con un adjetivo que representa la usabilidad percibida (ver Tabla 21). Dada esta información el puntaje obtenido se categoriza con el grado “B”, que se describe como bueno.

Tabla 21. Grado para puntajes SUS⁵⁸

Puntaje SUS	Grado	Adjetivo para la Calificación
> 80,3	A	Excelente
68 - 80,3	B	Bueno
68	C	Ok/Está bien
51 - 68	D	Pobre
< 51	F	Horrible

5.2.4. Consecuencia de las pruebas de usabilidad

Considerando 21 personas consultadas para realizar la prueba usabilidad, de los cuales 16 aceptaron, podemos concluir que el porcentaje de participación fue de 76,19%.

Consecuencia de las pruebas de usabilidad y retroalimentación por parte de los usuarios permitió reformular ciertas funcionalidades para dar solución a las necesidades expresadas por los usuarios, para así entregar una aplicación de mayor calidad.

A continuación, se listan los cambios realizados:

- Se consideró necesario una sección “Acerca de” en la aplicación para que así, los usuarios puedan comprender la idea detrás de MusycGo. Dicha información se muestra en la ventana principal.
- Por el hecho de que algunos usuarios consideraron que existían muchos elementos y colores diferentes en el mapa principal, se rediseñaron algunos marcadores y se quitaron otros, además de agregar un apartado que muestra la simbología del mapa.
- Se quitaron algunas animaciones en botones del mapa principal.
- Para que los usuarios entendieran la utilidad de haber iniciado sesión, se agregó una descripción en el apartado de “Ingresar Canción” y en los formularios de modificación.

Existen aspectos que no fueron modificados, p. ej., la obligación de escribir una justificativa para cada lugar (se considera que debe estar), no poder ingresar canciones directamente desde el mapa principal (existe otra sección dedicada a eso) y actualizar el mapa cuando se cambia una opción de los filtros de éste. Mientras que otros aspectos eran parte de ciertas restricciones de servicios externos, p. ej., reproductores de la vista previa de una canción tardaron en aparecer (son *iframe* que se conectan con los servidores de Spotify para que puedan aparecer) y también cuando las canciones se sobreponen al seleccionar más de una al mismo tiempo (Spotify no proporciona eventos observables para el botón “Reproducir”⁵⁹).

⁵⁸ <https://medium.com/uxpress/quantify-with-some-standardized-usability-tests-3a69f6746c63>

⁵⁹ <https://developer.spotify.com/documentation/widgets/guides/adding-a-spotify-play-button/>

6. CONCLUSIONES

Este proyecto permitió abordar y poner en práctica variadas habilidades obtenidas de la enseñanza universitaria, habilidades tanto técnicas, como comunicacionales y de análisis.

El desarrollo del presente proyecto conllevó a la desafiante tarea de investigar y aplicar el conocimiento necesario para ser un desarrollador que cubra la gran mayoría de las aristas del proyecto, un paso del largo camino a un perfil profesional *Full Stack*, aplicando componentes tales como:

- Entorno de servidor, redes y *hosting*.
- El modelo de datos.
- Lógica del negocio.
- Uso de API e interfaces.
- Experiencia de usuario.
- Entender las necesidades del cliente.

La innovación facilita a las personas de todo el mundo a conectarse con la cultura musical. La manera en que se accede a la música ha cambiado y no es la primera vez ni tampoco la última que habrá que reinventar la tecnología para la inclusión de la música, pero lo que es cierto es que la tecnología digital actual viene a servir como un poderoso medio de acceso universal, por lo tanto, hay que aprovecharlo.

Se comprobó (incluso con pruebas a usuarios) que la informática, el arte y la cultura pueden convivir en pro de la generación y difusión de conocimiento.

6.1. Cumplimiento de objetivos

Este proyecto de título comenzó con el objetivo general: “Fomentar la cultura musical regional, a través de una plataforma digital interactiva que permita crear una relación entre la localidad y la música”.

Este objetivo general se concluyó satisfactoriamente tras el cumplimiento de los siguientes objetivos específicos.

6.1.1. Revisar aplicaciones similares para realizar una interfaz interactiva

La revisión sistemática, la revisión de mercado y algunas aplicaciones web recomendadas por terceros, permitió conocer el estado actual de las tecnologías y métodos usados para abordar algunas de las características de MusycGo, algunas de las características de terceros fueron potenciadas para permitir una experiencia más interactiva.

6.1.2. Investigar metodología y arquitectura adecuada para este proyecto

Con la realización de una revisión sistemática y otra del mercado actual, sumadas a las investigaciones de *frameworks* de desarrollo se pudo asegurar una arquitectura probada y respaldada por la comunidad de desarrolladores.

6.1.3. Priorizar funcionalidades para realizar un prototipo en el menor tiempo posible

En conjunto con el profesor patrocinante se definieron los tipos de usuarios y las historias de usuarios, las cuales abordan una serie de funcionalidades que fueron priorizadas bajo el método MoSCoW, cumpliendo con el desarrollo e implementación de la totalidad de funcionalidades definidas como “*Must*” (“Debe ir”).

6.1.4. Validar la aplicación a través de pruebas y feedback de los usuarios

Durante el transcurso del proyecto se definió la realización de una prueba de usabilidad (SUS) a un grupo restringido de usuarios para que probaran el sistema y luego entregaran sus impresiones a través de un cuestionario. Los resultados permitieron afrontar y redefinir algunas funcionalidades que luego fueron implementadas, por lo tanto, se tomaron, se analizaron y se implementaron las opiniones y *feedback* de los usuarios.

6.1.5. Asegurar el tiempo requerido para el desarrollo de este proyecto, a través de la planificación y la estimación

A través de la metodología Scrum y el uso de plataformas de planificación se pudo asegurar una comunicación fluida y actualizada del estado del proyecto en todo momento, lo que permitió abordar un desarrollo de *software* complejo en un tiempo prudente de ciclos de análisis, desarrollo y pruebas.

6.1.6. Realizar diseño de la aplicación

Abordar el objetivo específico “Revisar aplicaciones similares para realizar una interfaz interactiva” y aplicando el conocimiento de la asignatura “Interfaz Humano-Computador” (HCI), permitieron diseñar y maquetar los módulos de MusycGo, además se aplicaron elementos, colores y distribuciones concordantes en toda la aplicación.

6.1.7. Difundir la aplicación para impactar a más usuarios

Se crearon y configuraron las redes sociales de MusycGo (Facebook, Instagram, YouTube y Twitter), con el fin de aprovechar su popularidad y la capacidad de alcance que tienen estas plataformas en la sociedad actual. Además, se diseñaron algunas figuras (logotipo, e icono) que entreguen identidad y pertenencia a MusycGo (ver Anexo D).

7. TRABAJOS FUTUROS

En primer lugar, tomando como base la cultura, se espera que MusycGo sea utilizado con ese fin, apoyar la cultura, las tradiciones y los estilos musicales. Lograr que las personas conozcan sus raíces, la música que los rodea y también, descubrir el gusto por la música y conocer su alrededor. Para ello se puede identificar e incluir a *stakeholders* que fomenten y estén comprometidos en este sentido, pensando en colaboraciones y apoyos. De momento puedo mencionar dos ejemplos: municipalidades (cultura y turismo) y establecimientos educacionales (educación, cultura, arte y tecnología).

Debido a que se encuentran creadas las cuentas de MusycGo en redes sociales (Facebook, Instagram, YouTube y Twitter) es factible diseñar una estrategia de *marketing* y posicionamiento digital, para publicar contenido, difundir y dar a conocer la plataforma.

Considerando que los datos están disponibles en Internet en un formato manejable (JSON) se espera que los interesados puedan realizar sus propios análisis de los lugares, canciones, artistas musicales, interacciones de los usuarios, trasfondos musicales, entre otros. Incluso, entre aquellos interesados se pueden generar investigaciones y colaboraciones multidisciplinarias que generen nuevas ideas o algún impacto no previsto.

Se debe considerar llevar el sistema a una aplicación móvil nativa, ya que puede ser favorable para ampliar servicios, funcionalidades y modelos de negocio. Tener la versión móvil permite aprovechar los sensores incorporados y almacenar datos de forma local para ofrecer servicios *offline*, con una posterior sincronización cuando se retome la conexión.

Se espera retomar y concluir aquellas historias de usuarios que no fueron abordadas en esta primera instancia, con el objetivo de entregar un producto más completo y refinado. Incluso, bajo la premisa de que ningún proyecto está terminado mientras exista el *Product Backlog* (metodología Scrum), siempre existirán o se descubrirán nuevas historias de usuario y características que puedan mejorar el proyecto y hacerlo más atractivo.

Técnicamente, se espera incluir el ingreso de canciones a través de nuevas plataformas de *streaming*, por ejemplo: YouTube, Deezer, SoundCloud, entre otros. Además, que estos puedan “convivir” en armonía dentro del sistema.

Para mejorar el rendimiento y la velocidad del intercambio de datos con el servidor, se podría incorporar la técnica del almacenamiento caché del lado del servidor (NodeJS) con la ayuda de ExpressJS.

Se considera una fuente fidedigna la inclusión directa de los propios artistas musicales, si bien, actualmente pueden colaborar, aún no existe una distinción formal para los que son artistas y los que no lo son, por lo que podrían colaborar y justificar su propia música sin tener duda con respecto a la veracidad, a fin de cuentas, ¿quién conoce mejor la música que los propios artistas que las idean, crean y componen?

8. REFERENCIAS

- AAM. (14 de Enero de 2018). *Estudio Música Chilena 2018 SCD*. Obtenido de AAM: https://aam.cl/wp-content/uploads/2019/01/gfk-scd-2018_final.pdf
- Aguirre A., F. (7 de Junio de 2017). *Usuarios chilenos de Spotify escuchan 52 millones de canciones a la semana*. Obtenido de La Tercera: <https://www.latercera.com/noticia/usuarios-chilenos-spotify-escuchan-52-millones-canciones-la-semana/>
- Angel, R., Camus, S., & Mansilla, C. (2008). *Plan de Apoyo técnico musical dirigido a los profesores de Educación General Básica, principalmente en NB1 y NB2*. Valparaíso: Universidad de Playa Ancha.
- Asociación Agencias de Medios. (Agosto de 2018). *Estudio Música Chilena 2018 SCD*. Obtenido de AAM: https://aam.cl/wp-content/uploads/2019/01/gfk-scd-2018_final.pdf
- Bertolini, P., Cabrera, F., & Ovalle, M. (4 de Enero de 2019). *Spotify y el impacto de las plataformas de streaming musical*. Obtenido de Medium: <https://medium.com/@streamusic12/spotify-y-el-impacto-de-las-plataformas-de-streaming-musical-ccc2f8016527>
- DB-Engines. (Febrero de 2020). *DB-Engines Ranking of Document Stores*. Obtenido de DB-Engines: <https://db-engines.com/en/ranking/document+store>
- Digital Equipment Corporation. (s.f.). *System Usability Scale*. Obtenido de measuringux: <https://www.measuringux.com/SUS.pdf> consultado en: 20-02-2020
- El País. (6 de Octubre de 2018). *Spotify cumple diez años*. Obtenido de El País: https://cincodias.elpais.com/cincodias/2018/10/06/companias/1538825256_585616.html
- El País. (9 de Octubre de 2006). *Google comprará la web YouTube por 1.650 millones de dólares*. Obtenido de El País: https://elpais.com/tecnologia/2006/10/09/actualidad/1160382485_850215.html
- ENO. (29 de Noviembre de 2019). *Trello. Qué es, para qué sirve y cómo funciona*. Obtenido de expertosnegociosonline: <https://www.expertosnegociosonline.com/que-es-trello-para-que-sirve/>
- Fowler, M. (1 de Mayo de 2006). *Continuous Integration*. Obtenido de Martin Fowler: <https://martinfowler.com/articles/continuousIntegration.html>
- GfK Adimark. (2018). *Chile Digital 2018*. Santiago: GfK.
- GfK Adimark. (2018). *Encuesta Música Chilena 2018*. Obtenido de GfK: https://www.gfk.com/fileadmin/user_upload/country_one_pager/CL/Edicion_Especial_CHILE3D_2018.pdf
- GfK. (Diciembre de 2018). *Chile Digital 2018*. Obtenido de GfK: https://www.gfk.com/fileadmin/user_upload/country_one_pager/CL/GfK_Chile_Digital_2018.pdf
- Hipertextual. (15 de Mayo de 2014). *¿Qué es una API?* Obtenido de Hipertextual: <https://hipertextual.com/archivo/2014/05/que-es-api/>
- IFPI. (2015). *Informe sobre la música digital de la IFPI 2015*. Obtenido de IFPI: <https://www.ifpi.org/downloads/Digital-Music-Report-2015-Spanish.pdf>

- IFPI. (24 de Septiembre de 2019). *IFPI releases Music Listening 2019*. Obtenido de IFPI: <https://www.ifpi.org/news/IFPI-releases-music-listening-2019>
- IFPI. (Septiembre de 2019). *Music Listening 2019*. Obtenido de ifpi: <https://www.ifpi.org/downloads/Music-Listening-2019.pdf>
- IMI Chile. (13 de Julio de 2018). *Estudio de Industria Musical en Chile destaca oportunidades en el mundo digital*. Obtenido de IMICHILE: <http://www.imichile.cl/2018/07/13/estudio-de-industria-musical-en-chile-destaca-oportunidades-en-el-mundo-digital/>
- IMI Chile. (13 de Julio de 2018). *MÚSICA CHILENA INDEPENDIENTE: OPORTUNIDADES Y NUEVAS EVIDENCIAS*. Obtenido de IMI Chile: http://www.imichile.cl/documentos/imi_oportunidades.pdf
- Infobae. (5 de Marzo de 2020). *Spotify Awards 2020: todos los ganadores*. Obtenido de Infobae: <https://www.infobae.com/america/entretenimiento/2020/03/06/spotify-awards-2020-todos-los-ganadores/>
- Kemp, S. (30 de Enero de 2020). *Digital 2020: 3.8 Billion People Use Social Media*. Obtenido de We Are Social: <https://wearesocial.com/blog/2020/01/digital-2020-3-8-billion-people-use-social-media>
- Linh Giang, N. A. (14 de Mayo de 2019). *Rank your product with standardized usability tests*. Obtenido de Medium: <https://medium.com/uxpress/quantify-with-some-standardized-usability-tests-3a69f6746c63>
- Measuring The User Experience. (s.f.). *System Usability Scale*. Obtenido de Measuring The User Experience: <https://www.measuringux.com/SUS.pdf> consultado en: 20-02-2020
- Mulligan, M. (5 de Diciembre de 2019). *Music Subscriber Market Shares H1 2019*. Obtenido de MIDIA: <https://www.midiaresearch.com/blog/music-subscriber-market-shares-h1-2019/>
- Pérez Porto, J., & Gardey, A. (2012). *Definición de música*. Obtenido de Definición.DE: <https://definicion.de/musica/>
- Spotify for Developers. (2020). *Authorization Guide*. Obtenido de Spotify for Developers: <https://developer.spotify.com/documentation/general/guides/authorization-guide/> consultado en: 20-02-2020
- Stack Overflow. (2018). *Developer Survey Results 2018*. Obtenido de Stack Overflow: <https://insights.stackoverflow.com/survey/2018#technology>
- Stack Overflow. (2019). *Developer Survey Results 2019*. Obtenido de Stack Overflow: <https://insights.stackoverflow.com/survey/2019#technology>
- Sutherland, J., & Schwaber, K. (Julio de 2013). *La Guía de Scrum*. Obtenido de Scrum Guides: <https://www.scrumguides.org/docs/scrumguide/v1/scrum-guide-es.pdf>
- Usability.gov. (2003). *System Usability Scale (SUS)*. Obtenido de Usability.gov: <https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html>

ANEXOS

Anexo A: Cuestionario aplicado a los usuarios en prueba de usabilidad

Usabilidad MusycGo

* Breve encuesta sobre el uso de la aplicación de georreferenciación musical MusycGo.

* Su respuesta es anónima.

* Algunas preguntas fueron formuladas en base al System Usability Scale (SUS).

***Obligatorio**

Ingrese su edad

Tu respuesta _____

¿Cuánto tiempo usó Musycgo? (tiempo estimado en minutos) *

Tu respuesta _____

¿Cuál crees que sería tu perfil de usuario más representativo dentro de la aplicación? *

Usuario Explorador (Ver, escuchar y aprender sobre los lugares que inspiraron canciones)

Usuario Colaborador (Ver, escuchar, modificar y aportar con nuevos lugares que inspiraron canciones)

No usaría la aplicación

Para cada afirmación (1 - 10) seleccione la opción que mejor represente su reacción al usar la aplicación.

1. Creo que me gustaría usar MusycGo con frecuencia. *

1 2 3 4 5

Totalmente en desacuerdo Totalmente de acuerdo

2. Encontré MusycGo innecesariamente complejo. *

1 2 3 4 5

Totalmente en desacuerdo Totalmente de acuerdo

3. Pensé que MusycGo era fácil de usar. *

1 2 3 4 5

Totalmente en desacuerdo Totalmente de acuerdo

4. Creo que necesitaría asistencia para poder usar MusycGo. *

1 2 3 4 5

Totalmente en desacuerdo Totalmente de acuerdo

5. Encontré que las diversas funciones en MusycGo estaban bien integradas. *

1 2 3 4 5

Totalmente en desacuerdo Totalmente de acuerdo

6. Pensé que había demasiada inconsistencia en MusycGo. *

1 2 3 4 5

Totalmente en desacuerdo Totalmente de acuerdo

7. Me imagino que la mayoría de la gente aprendería a usar MusycGo muy rápidamente. *

1 2 3 4 5

Totalmente en desacuerdo Totalmente de acuerdo

8. Me pareció muy engorroso / incómodo de usar. *

1 2 3 4 5

Totalmente en desacuerdo Totalmente de acuerdo

9. Me sentí muy seguro usando MusycGo. *

1 2 3 4 5

Totalmente en desacuerdo Totalmente de acuerdo

10. Necesitaba aprender muchas cosas antes de poder usar MusycGo. *

1 2 3 4 5

Totalmente en desacuerdo Totalmente de acuerdo

¿Cuál crees que sería la mejor manera de promocionar y/o divulgar la aplicación? *

Tu respuesta

¿Alguna funcionalidad no se comportó como esperaba?

Tu respuesta

¿Qué es lo que más te gustó y lo que menos te gustó de MusycGo?

Tu respuesta

¿En qué navegador ingresó a MusycGo? *

- Chrome
- Firefox
- Safari
- Opera
- Otro:

¿En qué dispositivo ingresó a MusycGo? *

- Móvil
- Computador
- Tablet
- Otro:

Proporcione cualquier comentario extra sobre MusycGo.

Tu respuesta

ENVIAR

Página 1 de 1

Nunca envíes contraseñas a través de Formularios de Google.

Anexo B: Resultados de la prueba SUS para cada pregunta por usuario

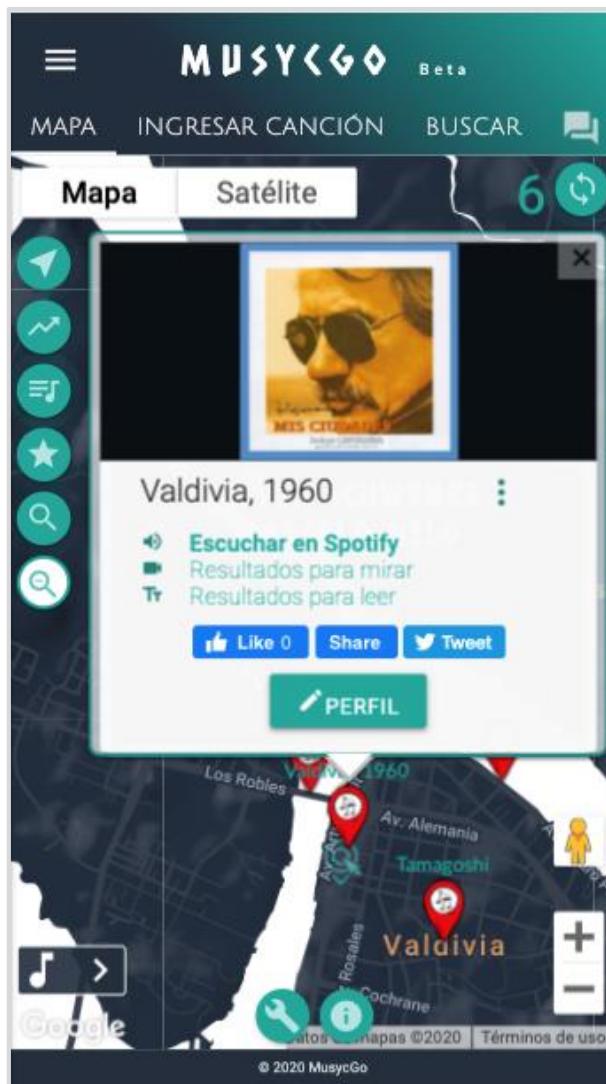
Usuario	P. 1	P. 2	P. 3	P. 4	P. 5	P. 6	P. 7	P. 8	P. 9	P. 10
1	4	1	3	3	5	2	3	2	5	2
2	4	2	4	1	4	1	3	1	5	2
3	3	1	5	4	3	2	4	2	3	1
4	3	3	4	3	4	2	4	2	4	3
5	4	1	4	2	3	2	4	1	4	2
6	4	2	4	1	4	3	4	2	3	1
7	3	1	5	1	4	1	5	2	4	1
8	4	2	4	1	3	4	4	2	5	1
9	4	2	3	1	3	2	2	3	3	2
10	4	2	5	1	2	2	4	1	1	1
11	3	1	4	1	3	2	4	2	3	1
12	4	3	5	2	4	1	4	3	2	2
13	4	2	2	2	4	1	4	1	1	1
14	3	3	4	4	4	1	3	2	4	2
15	4	3	3	3	4	3	3	4	4	3
16	3	5	4	3	1	2	4	5	3	1

Anexo C: Vista versión móvil – *responsive*

C. 1 Página principal



C. 2 Marcadores mapa principal



Anexo D: Figuras de MusycGo

D. 1 Icono



D. 2 Logotipo

MUSY<GO