

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

**DESARROLLO DE SISTEMA WEB Y APLICACIÓN MÓVIL
INFORMATIVA SOBRE MÚSICA NACIONAL ECUATORIANA**

DESARROLLO DE UN FRONTEND

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO SUPERIOR
EN DESARROLLO DE SOFTWARE**

DENISSE NOEMI CUMBAL GANCHOZO

DIRECTOR: ING BYRON LOARTE

DMQ, septiembre 2022

CERTIFICACIONES

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



Denisse Noemi Cumbal Ganchozo

denisse.cumbal@epn.edu.ec

denissecumbal2015@hotmail.com

Certifico que el presente trabajo de integración curricular fue desarrollado por Denisse Noemi Cumbal Ganchozo, bajo mi supervisión.



Ing. BYRON LOARTE, MSc.

DIRECTOR

byron.loarteb@epn.edu.ec

DECLARACIÓN DE AUTORÍA

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

Denisse Noemi Cumbal Ganchozo

DEDICATORIA

Dedico este trabajo curricular a mi Dios Jehová, por ayudarme con la fuerza y sabiduría necesaria para seguir adelante poniendo sus principios como pilares en mi vida y en mi educación, por ser mi guía en cada uno de mis pasos durante mi vida universitaria, a mi madre, por estar siempre ahí para darme ánimo cuando pensaba que no podía seguir adelante, a mi padre, quien con sus palabras de aliento hacían que recobre las fuerzas, a mis hermanas, por demostrarme el significado de la perseverancia al culminar sus estudios con éxito, a Juan Carlos, por estar siempre a mi lado y a mi abuelita Anita, por enseñarme que con un poco de sacrificio se puede lograr grandes cosas.

Denisse Noemi Cumbal Ganchozo

AGRADECIMIENTO

Agradezco a mis padres, hermanas, Abuelita y Juan Carlos por ser parte de mi vida y siempre apoyarme, animarme y brindarme su guía incondicional.

Un agradecimiento a la Escuela Politécnica Nacional, por haberme permitido ser parte de su familia y estudiar una carrera universitaria en ella, a cada uno de los profesores que conforman la Escuela de Formación de Tecnólogos, quienes con su paciencia y sabiduría han compartido sus conocimientos en toda mi etapa universitaria y con ello terminar con éxito este trabajo de titulación.

Denisse Noemi Cumbal Ganchozo

ÍNDICE DE CONTENIDO

CERTIFICACIONES.....	I
DECLARACIÓN DE AUTORÍA.....	II
DEDICATORIA.....	III
AGRADECIMIENTO.....	IV
ÍNDICE DE CONTENIDO.....	V
RESUMEN.....	VII
ABSTRACT.....	VIII
1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO.....	1
1.1 Objetivo general.....	1
1.2 Objetivos específicos.....	1
1.3 Alcance.....	2
1.4 Marco Teórico.....	3
2 METODOLOGÍA.....	6
2.1 Metodología de Desarrollo.....	6
Roles.....	6
Artefactos.....	7
2.2 Diseño de interfaces.....	11
Herramienta utilizada para el diseño.....	11
2.3 Diseño de la arquitectura.....	12
Patrón arquitectónico.....	12
2.4 Herramientas de desarrollo.....	13
3 RESULTADOS.....	15
3.1 <i>Sprint</i> 0. Configuración del ambiente de desarrollo.....	15
Descripción de requerimientos.....	15
Organización del proyecto <i>frontend</i>	15
Roles de usuarios.....	16
3.2 <i>Sprint</i> 1. Resultado de las interfaces para el usuario administrador y artista. 17	
Consumir varios métodos para iniciar sesión, cerrar sesión y recuperar contraseña del rol administrativo y artista.....	17
Consumir varios métodos para modificar el perfil del administrador y artista.....	19
Consumir varios métodos para gestionar las solicitudes de artistas.....	20
3.3 <i>Sprint</i> 3. Resultado de las interfaces para el usuario artista.	21

Consumir varios métodos para registrar el rol artista	21
Consumir varios métodos para enviar solicitud de afiliación	22
Consumir varios métodos para gestionar los géneros musicales	23
Consumir varios métodos para gestionar álbumes.....	24
Consumir varios métodos para gestionar canciones.	25
3.4 <i>Sprint 4</i>. Resultado de las interfaces para el usuario invitado.....	27
Buscar canciones.....	27
3.5 <i>Sprint 5</i> Pruebas y despliegue en el <i>frontend</i>	28
Aplicar y obtener resultados de pruebas unitarias	28
Aplicar y obtener resultados de pruebas de carga	29
Aplicar y obtener resultados de pruebas de compatibilidad.....	30
Aplicar y obtener resultados de pruebas de aceptación.....	31
Despliegue del <i>frontend</i>	32
4 Conclusiones	33
5 Recomendaciones	34
6 REFERENCIAS BIBLIOGRÁFICAS	35
7 ANEXOS	38
ANEXO I.....	39
ANEXO II.....	40
ANEXO III.....	88
ANEXO IV	89

RESUMEN

Actualmente en el Ecuador, no se cuenta con una plataforma de reproducción de música donde su enfoque primordial sea el de resaltar los géneros y a su vez impulsar el talento musical ecuatoriano, provocando que los artistas tengan que buscar plataformas internacionales, para dar a conocer sus versos en rimas y su talento en música, pero lamentablemente en estas plataformas no logran obtener un realce significativo, provocando que la identidad musical ecuatoriana frente a otro tipo de géneros se vean opacadas.

Con el objetivo de brindar un apoyo a la comunidad de artistas ecuatorianos con su música y sus letras que resaltan la belleza y sabiduría del pueblo ecuatoriano, en el presente proyecto se ha desarrollado un *frontend* que tiene como nombre (TakiTri), el cual permite la interacción directa del artista y su música por medio de la gestión de géneros, álbumes y canciones de forma rápida, sencilla y gratuita. Consiguiendo que las canciones de cada uno de ellos estén al alcance de todo el pueblo ecuatoriano.

El actual proyecto de Integración Curricular se encuentra estructurado de la siguiente forma: En primer lugar, se establece la problemática que existe en relación a la música nacional ecuatoriana y con ello determinar los objetivos, el alcance y el marco teórico respectivamente. En segundo lugar, se especifica la implementación de la metodología *Scrum* para el desarrollo del *frontend*, así como la realización de los diferentes prototipos, el uso de las herramientas, entre otros. En tercer lugar, se detalla cada una de las tareas que intervienen en cada uno de los módulos, así como los resultados que se han obtenido en cada *Sprint*. Por último, se especifica las conclusiones y posteriormente las recomendaciones que se han logrado durante el desarrollo de este proyecto de Integración Curricular.

PALABRAS CLAVE: *Frontend*, *Scrum*, identidad ecuatoriana, Gestión, Figma.

ABSTRACT

Currently in Ecuador, there is no music reproduction platform where its primary focus is standing out the genres and promoting the Ecuadorian musical talent, causing the artists to seek international platforms, to publicize their verses in rhymes and their music talent, but unfortunately, those platforms fail to obtain a highlighted meaning, causing the Ecuadorian musical identity against to genres are overshadowed.

In order to provide support to the Ecuadorian community artists with their music and lyrics that distinguish the beauty and wisdom of the Ecuadorian people, in this project we have developed a frontend called (TakiTri), which allows direct interaction of the artist and his music through genre management, albums and songs quickly, easily and free of charge. Getting that the songs of each of them are available to all Ecuadorians.

The current project of Curricular Integration is structured as follows: First, the problem that exists in relation to Ecuadorian national music is established and with this determines the objectives, scope and theoretical framework, respectively. Secondly, it specifies the implementation of the Scrum methodology for the development of the frontend, as well as the realization of the different prototypes, the use of the tools, among others. Thirdly, it details each of the tasks involved in each of the modules, as well as the results obtained in each Sprint. Finally, it specifies the conclusions and then the recommendations that have been achieved during the development of this Curricular Integration project.

KEYWORDS: Frontend, Scrum, Ecuadorian identity, Management, Figma.

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

En el Ecuador han surgido varios géneros musicales, que cuentan la historia de los antepasados indígenas, donde cada ritmo es parte de la identidad musical ecuatoriana, por ejemplo: el Pasillo refleja el espíritu del pueblo y es reconocido por el Patrimonio Cultural Inmaterial de la Humanidad como un distintivo musical ecuatoriano [1]. Por otra parte, el San Juanito como menciona Diario el Universo tiene un ritmo distintivo, originario de la provincia de Imbabura y el cual es otro distintivo musical ecuatoriano [2]. Sin embargo, en el país no existe una plataforma digital dedicada exclusivamente a los géneros ecuatorianos, en donde se realce la identidad musical ecuatoriana y en donde se dé la oportunidad a los artistas de colaborar de forma directa e indirecta en un sistema de reproducción de música. Logrando de esta manera, que los artistas opten por promover su contenido musical en plataformas internacionales como *Spotify*, *YouTube*, etc. [3] [4].

No obstante, en dichas plataformas internacionales los artistas ecuatorianos no tienen un lugar significativo en las mismas, provocando que la población del Ecuador no opte por escuchar sus canciones al momento de hacer uso de estas, por ejemplo, en el 2021 según el Diario Metro los géneros más escuchados en la plataforma de música *Spotify* por la población ecuatoriana correspondieron al Reguetón y otros géneros [5].

En el Ecuador existe el Derecho de propiedad Intelectual, la cual respalda el derecho de autor de cada artista ecuatoriano, el cual menciona “Los artistas, intérpretes y ejecutantes tienen el derecho de autorizar o prohibir la comunicación al público de sus interpretaciones o ejecuciones en directo, así como la fijación de sus interpretaciones y la reproducción de tales ejecuciones, por cualquier medio o procedimiento” [6].

Por lo antes mencionado, en el actual Trabajo de Integración Curricular se ha desarrollado un *frontend* llamado “TakiTri”, el cual pretende ser un espacio propio para la música ecuatoriana en donde los artistas den a conocer su música, géneros que resaltan la belleza de la mujer, álbumes que contribuyan a la identidad del pueblo y canciones que destaquen el talento del país. Logrando de esta manera que las actuales y nuevas generaciones tengan un espacio dedicado a la música nacional ecuatoriana gracias al uso de la tecnología.

1.1 Objetivo general

Desarrollar un *frontend* informativo sobre música nacional ecuatoriana.

1.2 Objetivos específicos

1. Levantar requerimientos para el *frontend* de música nacional ecuatoriana.

2. Diseñar el *frontend* en base a los requerimientos que se han obtenido.
3. Codificar los módulos del *frontend* en base a los requerimientos que se han obtenido
4. Probar el *frontend* y realizar el respectivo despliegue a producción para su difusión.

1.3 Alcance

El presente proyecto se contempla de un componente para desarrollar una aplicación del lado del cliente o comúnmente llamado *frontend*, el cual permite proporcionar al usuario la facilidad de realizar acciones de forma sencilla, esto es gracias al desarrollo web que se centra en trabajar correctamente con el consumo de datos, experiencia de usuario y la correcta integración con una serie de interfaces [7].

Es por esta razón que se ha desarrollado un *frontend* que se enfoca principalmente en proporcionar una experiencia de usuario significativa, correcto funcionamiento, adaptabilidad, integración y comunicación con un *backend* previamente desarrollado mediante el consumo de una serie de métodos para que toda la información sea visualizada en tiempo real y de forma organizada en cada una de las interfaces de usuario del *frontend* en los diferentes navegadores web, así como el gestionar el reproductor de música ecuatoriana. Utilizando para ello, una amplia gama de tecnologías de desarrollo actuales del lado del cliente, un patrón de arquitectura para una correcta organización a nivel de código, una metodología de desarrollo para el seguimiento y cumplimiento de los objetivos del proyecto y una serie de pruebas para garantizar la calidad del software [8]. Por último, el acceso a los datos se garantiza en su totalidad al contar con una serie de perfiles y acciones que se describen a continuación:

Perfiles dentro del *frontend*

- Administrador y Artista

Información general

- Consumir un método para presentar información en una página informativa.

Usuario administrador

- Consumir varios métodos para iniciar sesión, cerrar sesión y recuperar contraseña.
- Consumir varios métodos para modificar el perfil de usuario.
- Consumir varios métodos para gestionar las solicitudes de artistas.

Usuario artista

- Consumir varios métodos para iniciar sesión, cerrar sesión y recuperar contraseña.

- Consumir varios métodos para modificar el perfil de usuario.
- Consumir varios métodos para gestionar géneros musicales.
- Consumir varios métodos para gestionar álbumes.
- Consumir varios métodos para gestionar canciones.

1.4 Marco Teórico

En los últimos años la forma en que las personas disfrutan escuchar música ha ido cambiando radicalmente, siendo cada vez más fácil de realizar esta acción, esto es a causa de que la industria musical se ha digitalizado, haciendo que gran cantidad de canciones de distintos géneros y álbumes se encuentren en Internet [9], por ello al tener un *frontend* que proporcione al artista la oportunidad de llegar a sus fans de forma sencilla y rápida, abre sus posibilidades de seguir con su carrera musical.

Los servicios de *Streaming* de música, le dan al artista la libertad de gestionar su contenido, un control casi totalitario dentro de su carrera musical, especialmente a los artistas independientes, estos medios digitales les proporcionan la oportunidad de decidir como difundir su música, mediante el consumo de sus canciones en estos medios digitales, además de disminuir el plagio de las mismas según indica que los *Streaming* de música están causando que las personas no opten por la piratería al momento de escuchar música, ya que existen otras formas más sencillas y rápidas de realizar esta acción de forma gratuita [9].

Es por ello, que para llevar a cabo el desarrollo de un *frontend* es necesario utilizar varias tecnologías del lado del cliente, las cuales admiten la interacción del usuario con varios elementos como: menús, imágenes, botones, entre otros, donde la usabilidad permite una experiencia del usuario satisfactoria, para ello los principales lenguajes que permiten el desarrollo de aplicaciones del lado de *frontend* son: JavaScript, HTML y CSS pero no son los únicos pues también existen los *Frameworks* que se utilizan como plantillas para obtener un código más estructurado y ordenado como lo son Angular [10] y Vue [11] por citar algunos proporcionando estabilidad y escalabilidad a la misma cuando se requiera [7].

Las metodologías ágiles son técnicas utilizadas en el desarrollo del *software*, mediante el seguimiento de varios pasos se obtiene un sistema de forma alífera y configurable a las necesidades del consumidor, donde su prioridad es poder complacer al cliente con entregas continuas. Permitiendo de esta manera, responder de una forma rápida y organizada a los problemas que se presenten durante alguna de las etapas de la metodología [12].

Angular es un *Framework* que permite trabajar en una sola página, de esa forma se evita que la aplicación sea lenta o demore en cargar sus componentes. Además, integra de forma organizada los módulos que son necesarios dentro de un sistema como lo son vistas, rutas, controladores etc., logrando de esta manera tener una programación modular [13].

La metodología ágil *Scrum* permite tener un control de cada uno de los errores y resolverlos de forma rápida y segura en conjunto con el grupo de desarrolladores que reflejan el compromiso, el enfoque y el respeto; Para así lograr de esta manera cumplir con el objetivo del proyecto y seguimiento minucioso de cada fase del desarrollo [14].

Figma es una plataforma colaborativa para el diseño de interfaces pensada para que los desarrolladores de *software* puedan obtener código CSS, SVG y un primer acercamiento del sistema final mediante la creación de prototipos [15]. De esa forma se implementa cada aspecto solicitado por el dueño del producto de una forma rápida y sin tantas complicaciones, ya que esta plataforma permite comenzar un diseño desde cero o trabajar una plantilla acoplándose a la necesidad del usuario.

Un patrón de arquitectura dentro del desarrollo de un *frontend*, ayuda a prevenir incompatibilidades entre las herramientas en uso, ya que se analiza la comunican entre los equipos a utilizar, garantizando así el rendimiento de cada módulo en un lapso de tiempo corto. En donde su enfoque primordial es la implementación del MVC, las cuales se enfoca en 3 capas las cuales son: Lógica, Negocio e Interfaz [16].

Los métodos que proporciona una librería permiten hacer uso de técnicas en el desarrollo que colaboran entre sí para obtener un resultado favorable ante un objetivo propuesto [17], los cuales son una solución diseñada para cumplir con lo determinado dentro del proyecto en el menor tiempo posible.

Para el consumo e integración con el *backend* se hace por medio de una serie de métodos que ya están establecidos, configurados y testeados, similar como si se tratara de una API para que únicamente sea consumido por cualquier aplicación del lado del cliente o aplicación móvil sin importar el lenguaje de programación o arquitectura que se esté utilizando. Además, proporciona todo lo necesario para trabajar y gestionar con imágenes, archivos mp3, documentos y toda la información necesaria que necesita el *frontend* para poder presentar en cada una de sus interfaces de usuario sin dejar de lado la capa de seguridad [18].

Dentro del desarrollo *web* existen diferentes servicios que se acomodan a las necesidades de los clientes, los cuales permiten al proyecto que sea ágil y flexible. Esto es posible porque al construir un proyecto mediante servicios permite dividir un sistema en componentes que interactúan entre sí. Es por esta razón que al dividir el *software* en componentes *frontend* y

backend es eficiente y permite la integración de más aplicaciones de terceros sin causar inconvenientes a futuro [19].

La etapa de pruebas es importante dentro del desarrollo de *software*, ya que es un enfoque primordial para comprobar que la funcionalidad sea la correcta, es decir, que realice las actividades solicitadas como se ha estipulado en la etapa inicial de requerimientos. Por otra parte, es importante centrarse en esta etapa pues se puede prevenir y corregir todos los errores antes que salga a producción [20].

El despliegue dentro del desarrollo es una parte fundamental ya que el sistema de software está alojado en un servidor externo y no localmente, posibilitando que esté al alcance de los usuarios que tengan acceso a Internet [21].

2 METODOLOGÍA

Un estudio de casos es una indagación profunda que permite comprender de forma detallada la realidad de un tema determinado. El objetivo esencial es obtener la información real de la situación, por medio de la extracción de datos fundamentales que se adquieren en encuestas, entrevistas etc., y así lograr resultados prósperos para un buen análisis [22].

Correspondiente a esto, el presente Trabajo de Integración Curricular maneja un estudio de casos para la realización del componente específicamente el desarrollo del *frontend*, ya que se centra en el análisis de un tema determinado para su desarrollo con la ayuda de encuestas, entrevistas e indagación en otros sitios *web*, por lo cual se ha llegado a la conclusión que los artistas ecuatorianos no cuenta con una aplicación que les permita gestionar la música nacional de forma gratuita, en consecuencia a esto, se ha llevado a cabo el desarrollo de un *frontend* de música nacional ecuatoriana denominado "TakiTri" en donde los artistas ecuatorianos puedan gestionar desde una sola plataforma sus géneros, álbumes y canciones de forma gratuita gracias al uso de la tecnología y herramientas de desarrollo actuales.

2.1 Metodología de Desarrollo

Las metodologías de desarrollo permiten adaptarse a las necesidades del cliente dando solución y aceptación positiva a los cambios que puedan surgir durante el proceso de desarrollo de *software* y reflejando un servicio de calidad en periodos de tiempo relativamente cortos [23].

En la actualidad el uso de metodologías ágiles, permite tener un mayor control en cada *Sprint* que se planifica, realiza y entrega, es por ello, que en el desarrollo del *frontend* se ha implementado la metodología ágil *Scrum*, para llevar un control adecuado de cada una de las iteraciones que se han planificado realizar y entregar paulatinamente. Además, esta metodología ágil tiene un conjunto de artefactos y roles los cuales permiten guiar de manera satisfactoria todo el proceso de desarrollo de *software* desde un enfoque totalmente ágil e iterativo [12].

Roles

Scrum permite incorporar varios roles con diferentes actividades dentro de un equipo de trabajo, lo cual permite que el proyecto se lo realice de forma eficiente en un lapso de tiempo relativamente corto. Por otra parte, cada uno de estos roles permiten obtener un producto final de calidad [24], a continuación, se presentan cada uno de los roles que han sido participes en el desarrollo del *frontend*.

Product Owner

Este rol se encarga de establecer los elementos del componente a realizar y que los miembros del equipo comprendan lo solicitado. Además, la persona asignada con este rol toma todas las decisiones antes del desarrollo del producto *software* y dependiendo de los objetivos a cumplir [24]. En ese sentido, la **TABLA I** presenta a la persona asignada a este rol la cual posee todas las habilidades necesarias para desempeñar dicho rol mencionado anteriormente.

Scrum Master

Este rol al ser el líder del equipo de desarrollo se encarga de supervisar que todo el proceso *Scrum* se lo realice de la forma correcta, además de que se esté aplicando las buenas prácticas en cada implementación del producto *software*, pues el desempeño correcto de este rol se refleja en consecuencia en la última etapa de la fase de la metodología y de la aprobación del producto final por parte del *Product Owner* [24]. En ese sentido, en la **TABLA I** se presenta a la persona asignada a este rol para desempeñar las funcionalidades anteriormente mencionadas, mediante la ejecución de reuniones semanales y una buena comunicación con todo el equipo de trabajo.

Development Team

Este rol generalmente está formado por un grupo de 3 a 5 desarrolladores, que de forma organizada cada uno de ellos sabe cuáles son sus actividades a realizar, para lograr de esta manera obtener un incremento en cada iteración del desarrollo [24]. Por consiguiente, en la **TABLA I** se visualiza el rol que se ha desempeñado.

TABLA I: Designación de Roles.

ROLES	NOMBRES
<i>Product Owner</i>	Ing. Byron Loarte, MSc.
<i>Scrum Master</i>	Ing. Byron Loarte, MSc.
<i>Development Team</i>	Denisse Cumbal

Artefactos

Los artefactos dentro de *Scrum* destacan los efectos de una excelente implementación de la metodología para poder garantizar de esa manera el rendimiento y eficacia del proyecto en cada *Sprint* entregado al usuario [24]. A continuación, se presenta cada uno de los artefactos

que se han seleccionado para el desarrollo del *frontend* y de esta manera garanticen el correcto registro de datos e información.

Recopilación de Requerimientos

Dentro del desarrollo del *frontend*, la recopilación de requerimientos ayuda a tener un punto de referencia para las actividades que se realizan y también estimar tiempos, recursos, cronograma de entregables que se llevan a cabo durante todo el desarrollo del proyecto. Por otra parte, permite evaluar el avance y cumplimiento del mismo [24].

La Recopilación de requerimientos para el desarrollo del *frontend* se llevan a cabo mediante una encuesta con el dueño del producto, en donde la información que se ha obtenido se refleja en la implementación y diseño de los diferentes módulos. Prueba de ello, la **TABLA II**: Requerimientos que se han recopilado se visualiza el formato que se ha utilizado para la elaboración de la recopilación de requerimientos. No obstante, la tabla completa con toda la información más profunda y detallada se encuentra en el **ANEXO II** del actual documento.

TABLA II: Requerimientos que se han recopilado.

RECOPIACIÓN DE REQUERIMIENTOS		
TIPO DEL SISTEMA	ID - RR	ENUNCIADO DEL ÍTEM
FRONTEND	RR001	El usuario con perfil invitado, artista y administrador tienen la posibilidad de visualizar una página informativa.

Historias de Usuario

Se utilizan para plasmar los requerimientos de los clientes de una forma sintetizada, corta y fácil de entender para el equipo de desarrollo, en donde se especifica que funcionalidad va a tener el producto o servicio a desarrollar. Además, estas Historias deben ser sencillas de recordar, entender y de implementarse en un periodo de tiempo determinado [24]. En consecuencia, a lo citado anteriormente cada una de las Historias de Usuario representa una de las diferentes funcionalidades del *frontend* es por ello que en la **TABLA III** se presenta un ejemplar de las Historias de Usuario que se ha realizado bajo los parámetros ya mencionados, sin embargo, las demás Historias se encuentran en el **ANEXO II** del presente documento.

TABLA III: Historia de usuario 01 - Visualizar una página informativa.

HISTORIA DE USUARIO	
Identificador: HU001	Usuario: Invitado
Nombre historia: Visualizar una página informativa	
Prioridad en Negocio: Media	Riesgo en Negocio: Media
Iteración asignada: 2	
Responsable: Denisse Cumbal	
<p>Descripción: El usuario invitado en el <i>frontend</i> pueden visualizar la página informativa, además de interactuar en el menú principal con la siguiente información:</p> <ul style="list-style-type: none"> • Inicio • Misión y Visión • Programadora 	
<p>Observación:</p> <p>Todos los perfiles pueden visualizar la página informativa, pero necesitan iniciar sesión en la misma para tener acceso a módulos distintos dependiendo el rol correspondiente.</p>	

Product Backlog

Es un conjunto de todos los requerimientos extraídos en donde su finalidad es poder maximizar la aplicación dentro de la etapa de desarrollo. Además, dentro del contenido de este *Product Backlog* se encuentran las tareas principales y secundarias dependiendo su prioridad en el desarrollo y el estado de cada una de estas tareas [14]. En ese aspecto, la **TABLA IV** presenta el formato que se ha realizado para plasmar los requerimientos que se han solicitado en base a la prioridad del negocio, por otra parte, la tabla completa del *Product Backlog* se encuentra en el **ANEXO II** del actual documento.

TABLA IV: *Product Backlog*.

ELABORACIÓN DEL <i>PRODUCT BACKLOG</i>				
ID – HU	HISTORIA DE USUARIO	ITERACIÓN	ESTADO	PRIORIDAD

HU001	Visualizar una página informativa	1	Finalizada	Media
HU002	Iniciar sesión, cerrar sesión y recuperar contraseña	1	Finalizada	Alta

Sprint Backlog

Es una tabla donde se encuentran todas las iteraciones del producto a desarrollar dependiendo de la prioridad del cliente y permitiendo de esta manera ver paso a paso el detalle de cada una de las tareas pendientes [14]. Por lo tanto, la **TABLA V** muestra el formato que se ha utilizado para elaborar el *Sprint Backlog* mediante en el cual se lista 5 *Sprints* que son: Configuración del ambiente de desarrollo, información principal del *frontend*, recuperación de contraseña del usuario, modificar el perfil administrador; solicitudes de artistas, registro de perfil artista, géneros musicales, álbumes, canciones, pruebas y despliegue del *frontend*. Por último, la tabla completa con el detalle de los demás *Sprints* se encuentra en el **ANEXO II** del actual documento.

TABLA V: *Sprint Backlog*.

ELABORACIÓN DEL <i>SPRINT BACKLOG</i>						
ID-SB	NOMBRE	MÓDULO	ID-HU	HISTORIA DE USUARIO	TAREAS	TIEMPO ESTIMADO
SB000	Configuración del ambiente de desarrollo	-----	-----	-----	<ul style="list-style-type: none"> Definir los requerimientos funcionales y no funcionales del <i>frontend</i> Configuración de las 	20 horas

					herramientas del <i>frontend</i> <ul style="list-style-type: none"> • Configuración del ambiente de desarrollo del <i>frontend</i> 	
--	--	--	--	--	---	--

2.2 Diseño de interfaces

Es un boceto que se crea en base a los requerimientos del dueño del producto, es decir es una parte visual en donde el usuario puede interactuar de mejor forma con los colores, estilos y demás elementos que va a tener el producto una vez que se empiece la etapa de codificación, logrando de esta manera tener una primera aceptación por parte del cliente y corregir posibles errores o fallas a nivel de interactividad o experiencia de usuario [25].

Herramienta utilizada para el diseño

A través de una herramienta de diseño enfocada principalmente en el desarrollo de interfaces como es Figma se procede a crear los diferentes prototipos de cada uno de los módulos que corresponden al *frontend*, los cuales toman realce en la fase de desarrollo. Prueba de ello la **Fig. 1**, muestra un primer prototipo con la información general, la cual está compuesta por una serie de elementos como: *header*, *footer*, *navbar*, *cards* entre otros. Por último, los demás prototipos que se han diseñado por cada módulo se encuentran en el **ANEXO II** del actual documento.

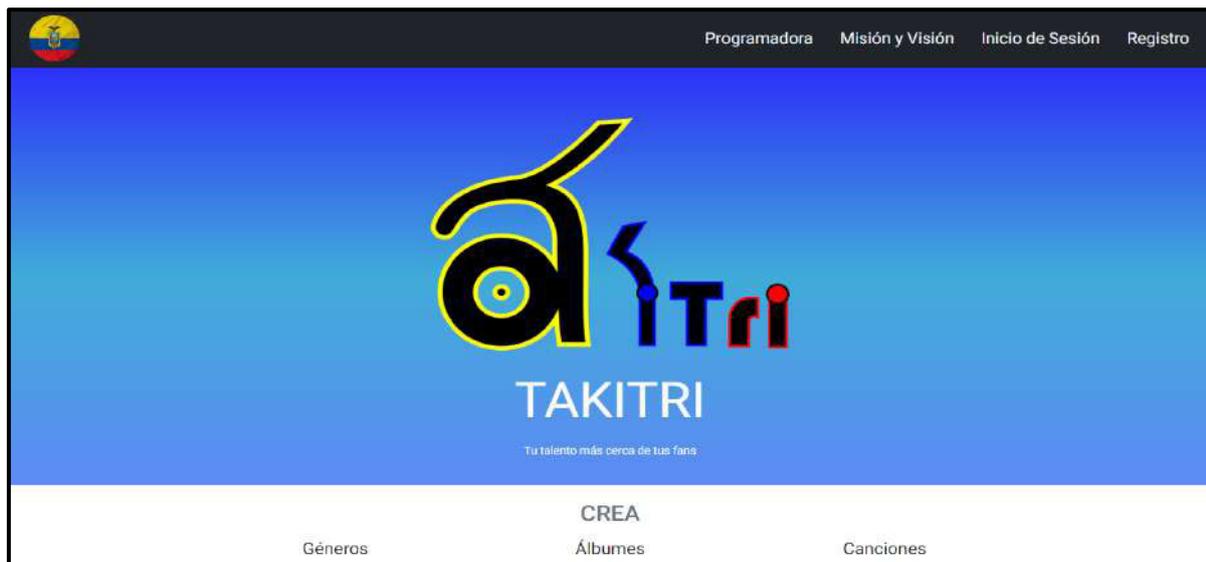


Fig. 1: Prototipo para mostrar la información general.

2.3 Diseño de la arquitectura

Son las diferentes interacciones que existen entre los componentes internos de un *software* que permita un buen funcionamiento y escalabilidad a futuro e integración con otras herramientas y librerías [25]. Por consiguiente, en el desarrollo del *frontend* se ha implementado un diseño arquitectónico el cual se detalla a continuación.

Patrón arquitectónico

El patrón arquitectónico MVC comúnmente llamado Modelo, Vista y Controlador se encarga de constituir y organizar los diferentes componentes internos de un *software*, separando la lógica, interfaz e información del mismo, que dentro del desarrollo es un modelo a seguir y que en muchos sistemas o aplicaciones de gran escala lo integran como parte de su núcleo, posibilitando de esta manera tener un control mucho más ordenado y estructurado en cada una de las capas del sistema *software* [26].

- **Modelo:** es la capa que gestiona todos los datos que va a interactuar con el sistema, especialmente la parte de la lógica.
- **Vista:** es la capa donde se presenta el contenido al usuario y toda la información de forma visual.
- **Controlador:** es la capa que se encarga de gestionar la información entrante por parte del usuario siendo el mediador entre el modelo y la vista.

En ese sentido, se procede a mostrar el patrón arquitectico que se ha implementado para el desarrollo del *frontend* como se visualiza en la **Fig. 2**, lo cual posibilita una alta compatibilidad con todas las librerías y herramientas de desarrollo, pruebas y despliegue.

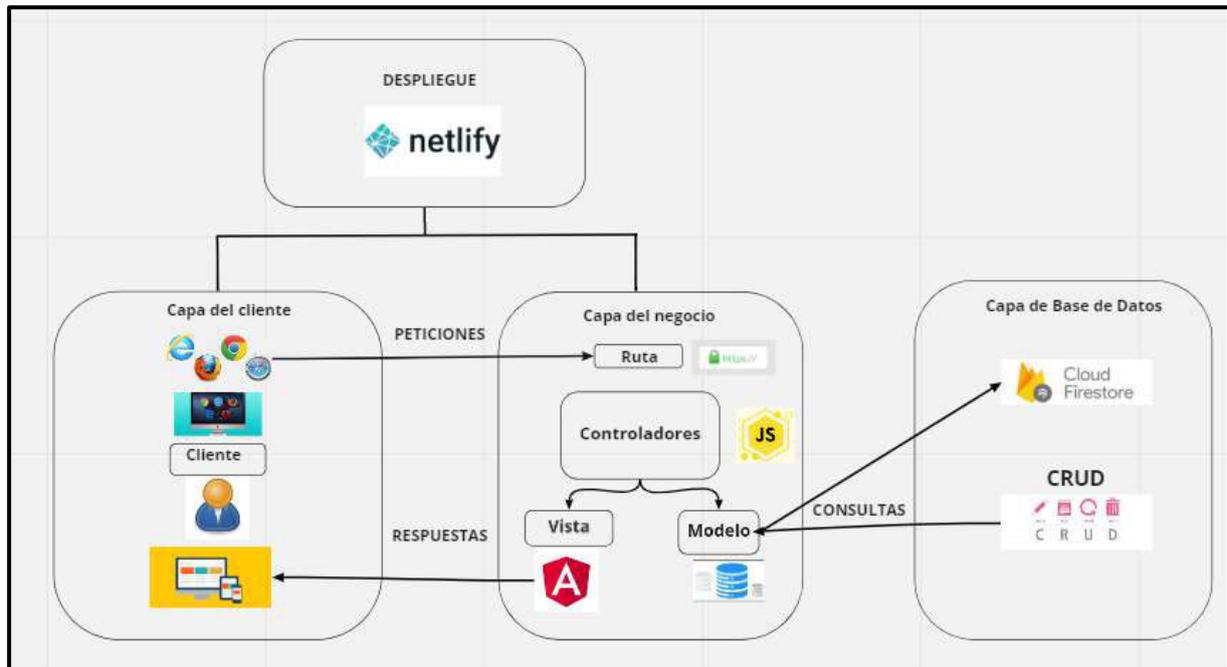


Fig. 2: Patrón arquitectónico – frontend.

2.4 Herramientas de desarrollo

Las herramientas de desarrollo para el presente proyecto son seleccionadas en base a cada uno de los requerimientos, haciendo óptimas para el desarrollo del mismo y que sean compatibles entre ellas, las cuales se describen a continuación por medio de la **TABLA VI**, la cual además muestra una breve justificación de su implementación.

TABLA VI: Herramientas para el desarrollo del frontend.

HERRAMIENTA	JUSTIFICACIÓN
Angular	Angular es un <i>Framework</i> que permite crear aplicaciones web de una sola página (SPA) mediante el uso de <i>TypeScript</i> y una estructura bien definida [10].

Angular CLI	Es un CLI o intérprete de línea de comando la cual colabora con Angular en la creación de proyectos y demás componentes por medio de una serie de comandos preestablecidos [27]
Angular Material	Es un conjunto de módulos y componentes visuales creados primordialmente para la parte visual de Angular, los mismos que permiten que sean adaptables a cualquier tamaño del dispositivo [28].

Librerías

Dentro del desarrollo del *frontend*, existen diferentes librerías las cuales aportan una serie de ventajas para que el proyecto sea más eficiente, evitando errores al momento de la compilación del mismo y ahorrando tiempo en cada *Sprint* que se ha planificado [31]. La **TABLA VII**, muestra una breve justificación de todas las librerías que se han utilizado.

TABLA VII: Librerías para el desarrollo del *frontend*.

LIBRERÍA	DESCRIPCIÓN
<i>material-app</i>	Librería para el diseño de las interfaces del <i>frontend</i> [28].
<i>mat-icon</i>	Librería para mostrar íconos en Angular [29].
<i>sweetalert2</i>	Librería para mostrar las alertas o mensajes al usuario de una forma sencilla [30].

3 RESULTADOS

A continuación, se presentan a detalle los resultados que se han alcanzado en cada módulo y elementos visibles que se han realizado para el componente del *frontend*, además de las diferentes pruebas que se han realizado y el respectivo despliegue del mismo. Todos los resultados que se han alcanzado se los muestran por medio de *Sprints*, los cuales se han planificado con anterioridad como se detalla en el **ANEXO II** del actual documento.

3.1 *Sprint 0*. Configuración del ambiente de desarrollo

Las tareas para el *Sprint 0* tienen que ver con:

- Descripción de requerimientos.
- Organización del proyecto *frontend*.
- Roles de usuarios.

Descripción de requerimientos

Visualizar página informativa

En el *frontend* los usuarios con perfil invitado, artista y administrador pueden visualizar la página informativa del componente que son: Programadora, Misión, Visión e Inicio. Es importante mencionar que para visualizar esta página el usuario independiente de su rol no debe iniciar sesión en el *frontend*.

Organización del proyecto *frontend*

La herramienta *Visual Studio Code* se usa como entorno de desarrollo del componente *frontend*, la cual colabora con la estructura de los diferentes módulos, directorios y archivos en correlación con la arquitectura que se ha mostrado de antemano, es por ello que en la **Fig. 3** se muestra la organización para el desarrollo.

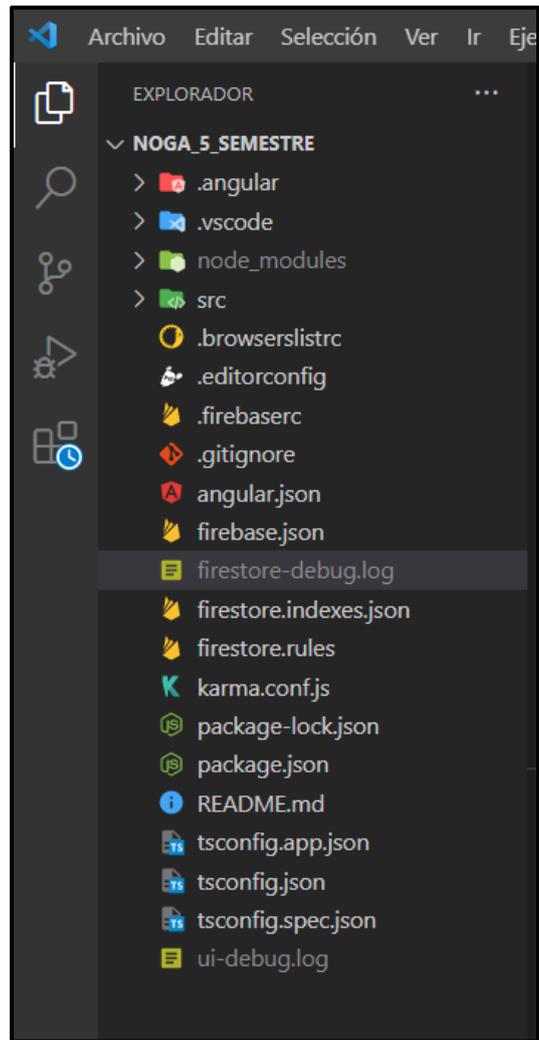


Fig. 3: Organización del *frontend*.

Roles de usuarios

A continuación la **Fig. 4** muestra los diferentes roles de cada usuario dentro del *frontend* los cuales son: administrador, artista e invitado. Al momento de iniciar sesión cada rol tiene acceso a módulos distintos, menos el rol invitado.

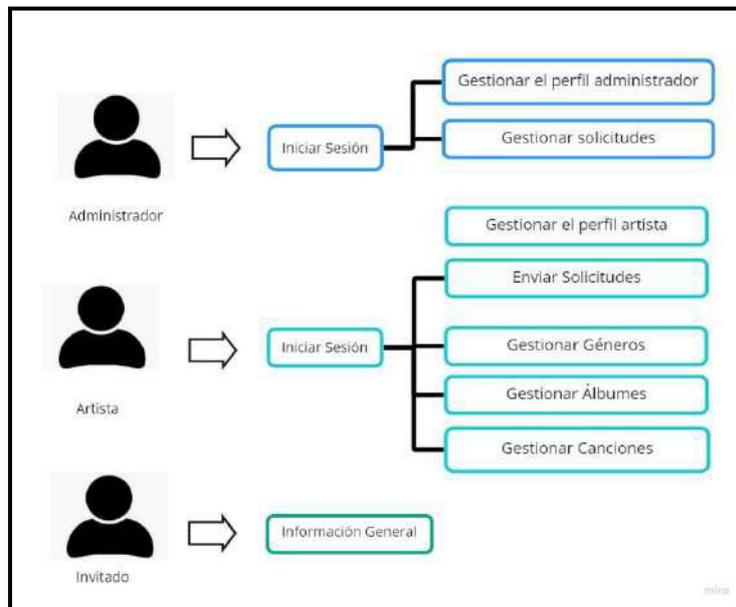


Fig. 4: Roles de usuario para el *frontend*.

3.2 *Sprint* 1. Resultado de las interfaces para el usuario administrador y artista.

Las tareas para el *Sprint* 1 están conformado por:

- Consumir varios métodos para iniciar sesión, cerrar sesión y recuperar contraseña del rol administrativo y artista.
- Consumir varios métodos para modificar el perfil del administrador y artista.
- Consumir varios métodos para gestionar las solicitudes de artistas.

Consumir varios métodos para iniciar sesión, cerrar sesión y recuperar contraseña del rol administrativo y artista

En el *frontend* el usuario con perfil administrativo y artista tiene la posibilidad de iniciar sesión mediante un correo y contraseña generada y otorgada por el *backend* en el caso del perfil administrador, pero para el perfil artista ingresa con sus credenciales generadas por su autoría, además una vez dentro del *frontend* puede cerrar sesión, en el caso que el usuario administrador y artista se olviden sus credenciales tienen la opción de recuperar su cuenta mediante un formulario en donde se solicita el ingreso del correo electrónico. Todo lo mencionado anteriormente es posible gracias al consumo de métodos por parte del *backend* y que es presentado en los componentes visuales del *frontend* como se puede ver en las **Fig. 5**, **Fig. 6** y **Fig. 7**. Por otra parte, se detalla el funcionamiento completo que se puede ver de mejor manera en el **ANEXO III** del actual proyecto.

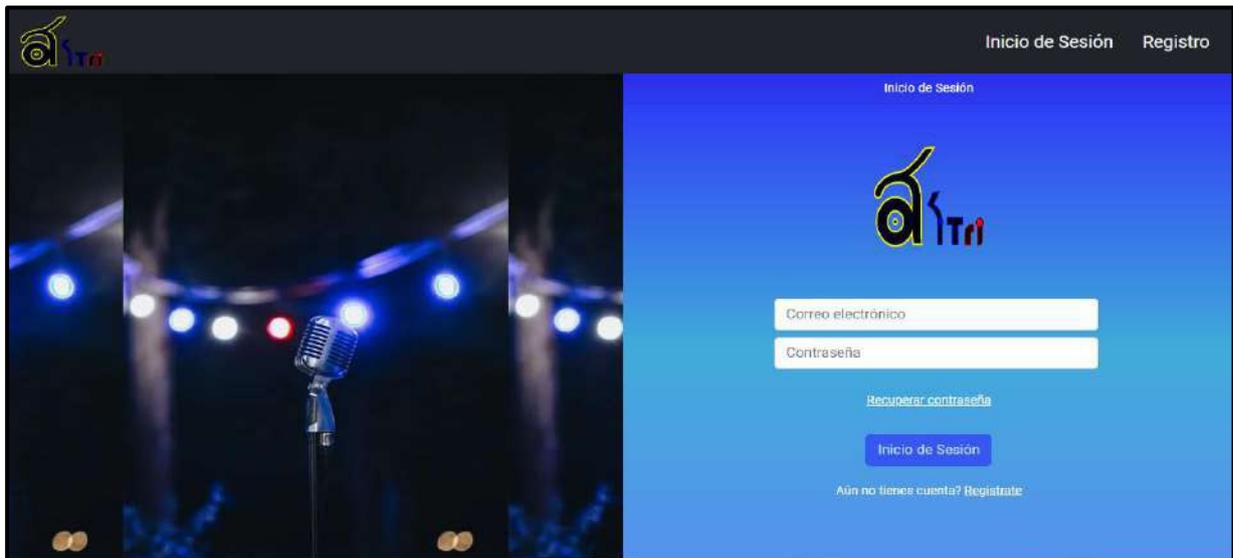


Fig. 5: Ingreso de usuarios administrador y artista.

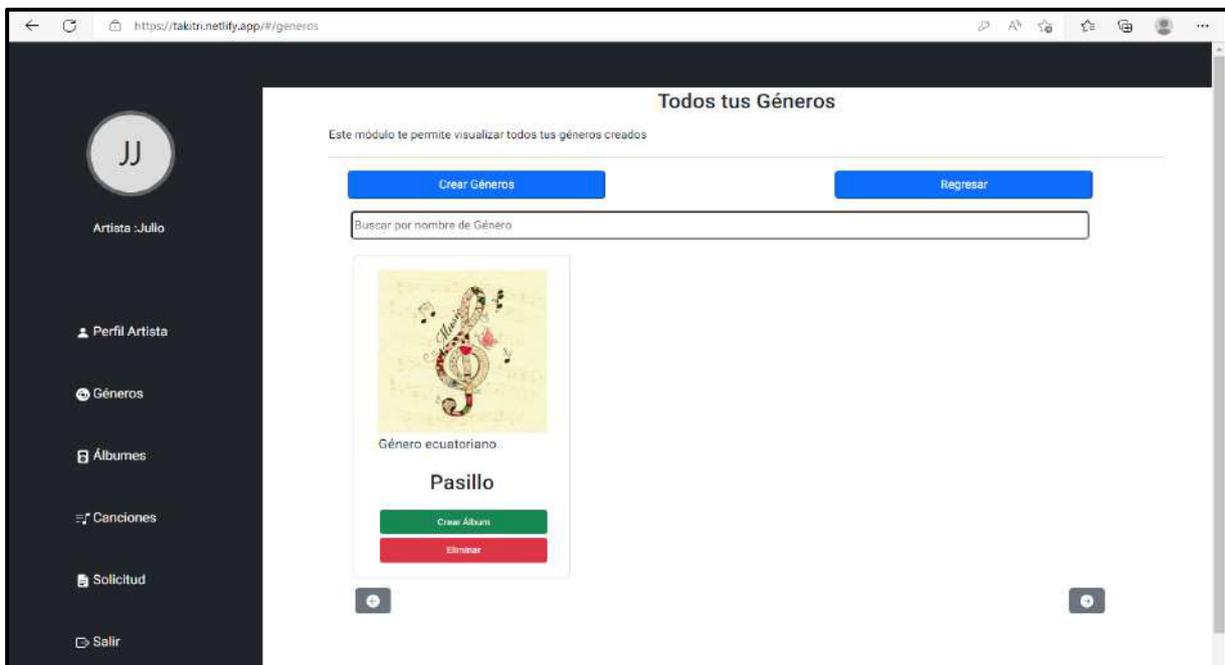


Fig. 6: Cierre de sesión de usuarios administrador y artista.

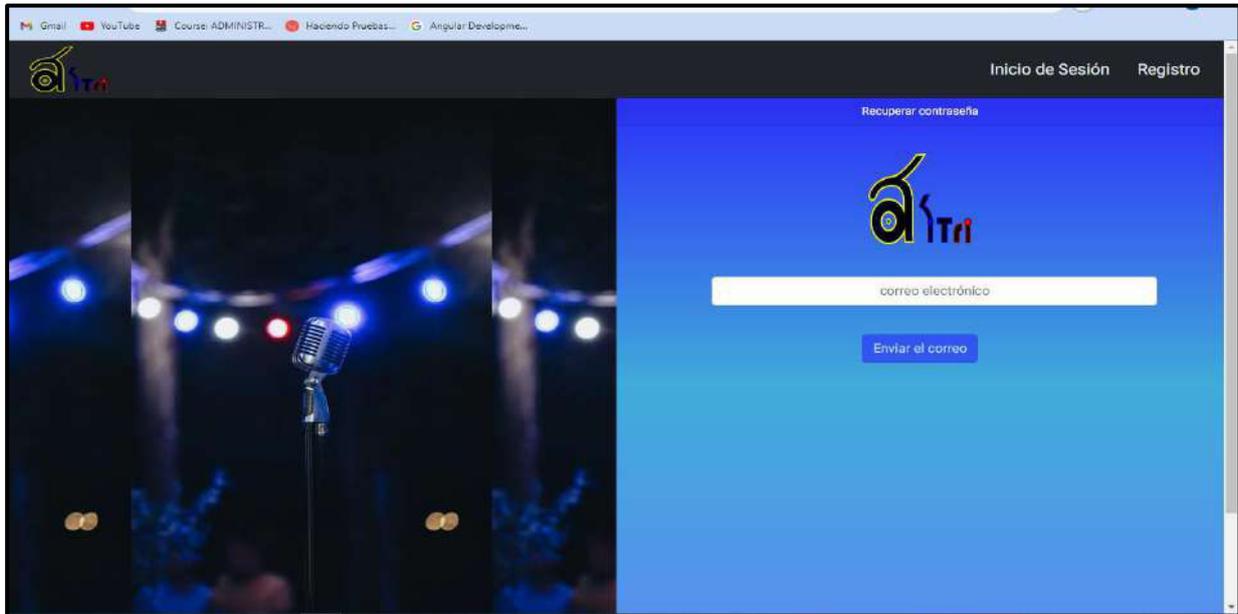


Fig. 7: Recuperar contraseña de usuarios administrador y artista.

Consumir varios métodos para modificar el perfil del administrador y artista.

En el *frontend* el usuario con perfil administrador y artista tienen la posibilidad de visualizar su perfil que es: nombre, apellido, correo electrónico, alias y una imagen conformada de su nombre y apellido, además pueden editar su perfil en los campos seleccionados como lo es: nombre, apellido, alias, entre otros. Todo esto es posible gracias al consumo de varios métodos por parte del *backend*, que es presentado en los componentes visuales del *frontend* como se puede ver en la **Fig. 8**. Por otra parte, se detalla el funcionamiento completo que se puede ver de mejor manera en el **ANEXO III** del actual documento.

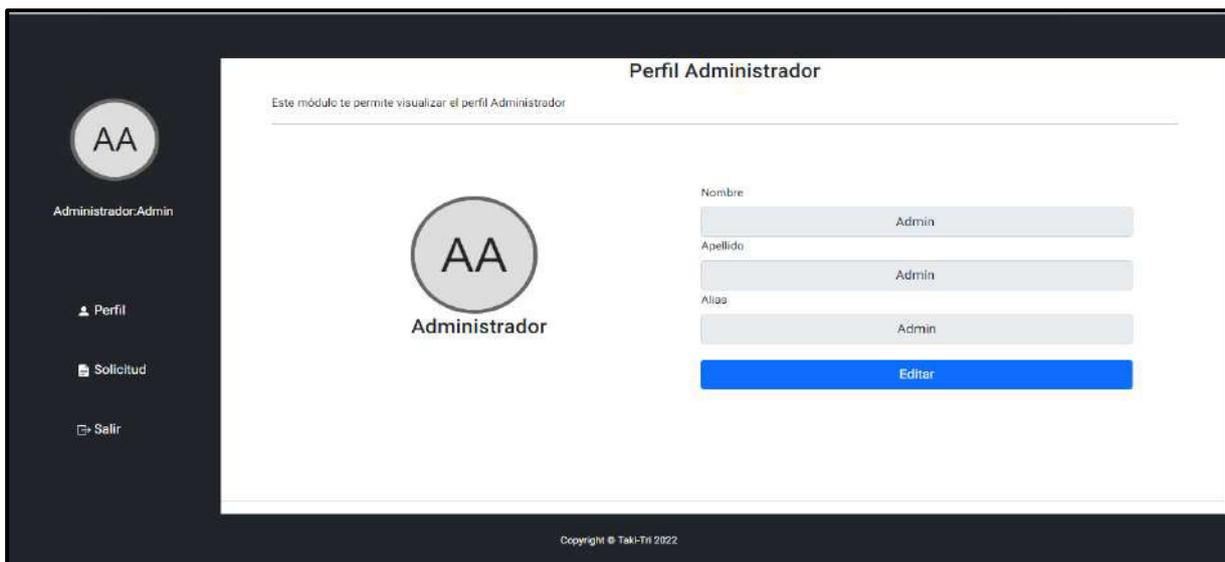
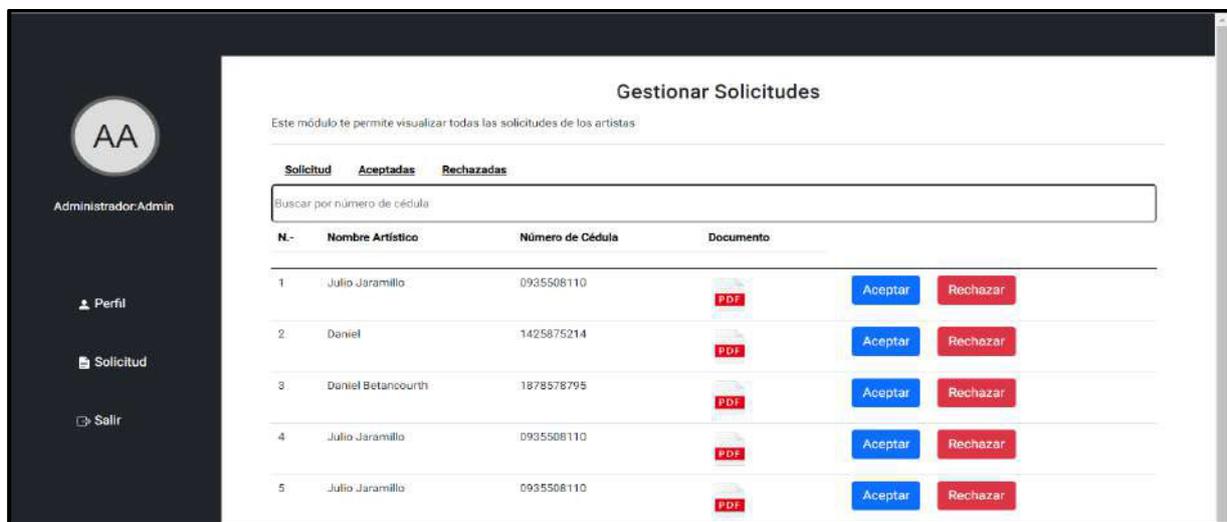


Fig. 8: Modificar el perfil del usuario administrador.

Consumir varios métodos para gestionar las solicitudes de artistas

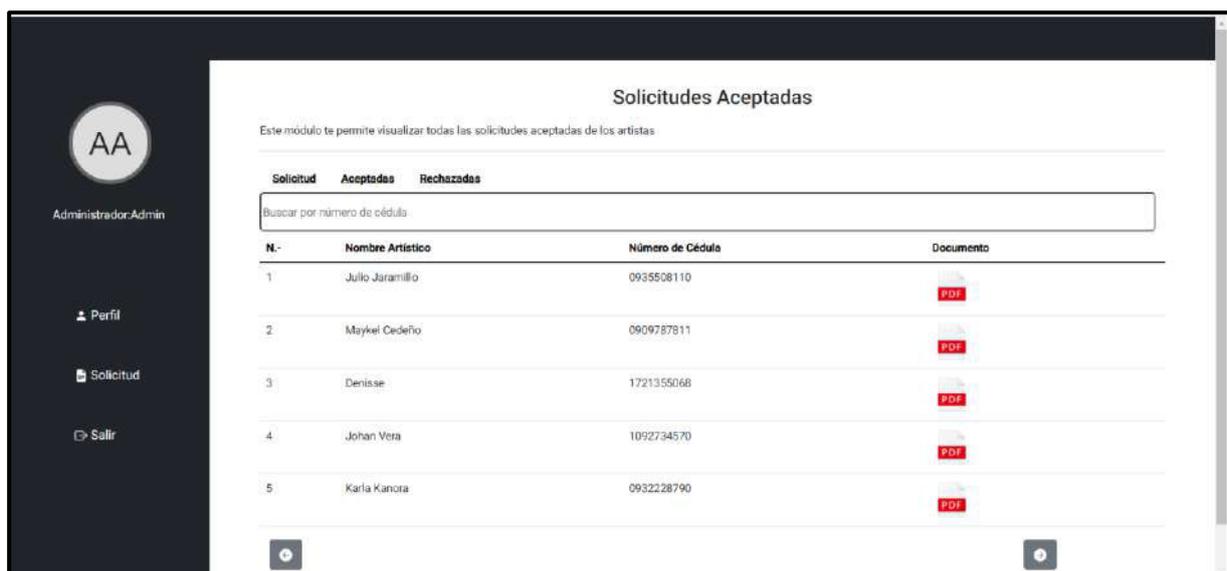
En el *frontend* el usuario con perfil administrador tiene la posibilidad de visualizar las solicitudes de los usuarios artistas que desean ser parte de “TakiTri”, para poder analizar si son aceptadas o rechazadas, es importante mencionar que las solicitudes son aprobadas si cuentan con un documento para su validación y aprobación, caso contrario el administrador puede rechazar dicha solicitud, explicando porque fue rechazada y enviado al usuario artista un mensaje de la razón. Todo esto es posible gracias al consumo de varios métodos por parte del *backend* que es presentado en los componentes visuales del *frontend* como se puede ver en las **Fig. 9, Fig.10, Fig.11**. Por otra parte, se detalla el funcionamiento completo que se puede ver de mejor manera en el **ANEXO III** del actual documento.



The screenshot shows the 'Gestionar Solicitudes' (Manage Requests) interface. It features a sidebar with a user profile 'Administrador:Admin' and navigation options for 'Perfil', 'Solicitud', and 'Salir'. The main content area has a title 'Gestionar Solicitudes' and a subtitle 'Este módulo te permite visualizar todas las solicitudes de los artistas:'. Below this is a search bar labeled 'Buscar por número de cédula'. A table displays five requests, each with a 'N.-' (ID), 'Nombre Artístico' (Artist Name), 'Número de Cédula' (ID Number), and 'Documento' (Document) column. The 'Documento' column contains PDF icons. To the right of each row are two buttons: 'Aceptar' (Accept) in blue and 'Rechazar' (Reject) in red.

N.-	Nombre Artístico	Número de Cédula	Documento		
1	Julio Jaramillo	0935508110	PDF	Aceptar	Rechazar
2	Daniel	1425875214	PDF	Aceptar	Rechazar
3	Daniel Betancourth	1878578795	PDF	Aceptar	Rechazar
4	Julio Jaramillo	0935508110	PDF	Aceptar	Rechazar
5	Julio Jaramillo	0935508110	PDF	Aceptar	Rechazar

Fig. 9: Nuevas solicitudes de artistas.



The screenshot shows the 'Solicitudes Aceptadas' (Accepted Requests) interface. It features the same sidebar as Figure 9. The main content area has a title 'Solicitudes Aceptadas' and a subtitle 'Este módulo te permite visualizar todas las solicitudes aceptadas de los artistas:'. Below this is a search bar labeled 'Buscar por número de cédula'. A table displays five accepted requests, each with a 'N.-' (ID), 'Nombre Artístico' (Artist Name), 'Número de Cédula' (ID Number), and 'Documento' (Document) column. The 'Documento' column contains PDF icons. There are no action buttons for these requests.

N.-	Nombre Artístico	Número de Cédula	Documento
1	Julio Jaramillo	0935508110	PDF
2	Maykel Cedeño	0909787811	PDF
3	Denisse	1721359068	PDF
4	Johan Vera	1092734570	PDF
5	Karla Kanora	0932228790	PDF

Fig.10: Solicitudes aceptadas.

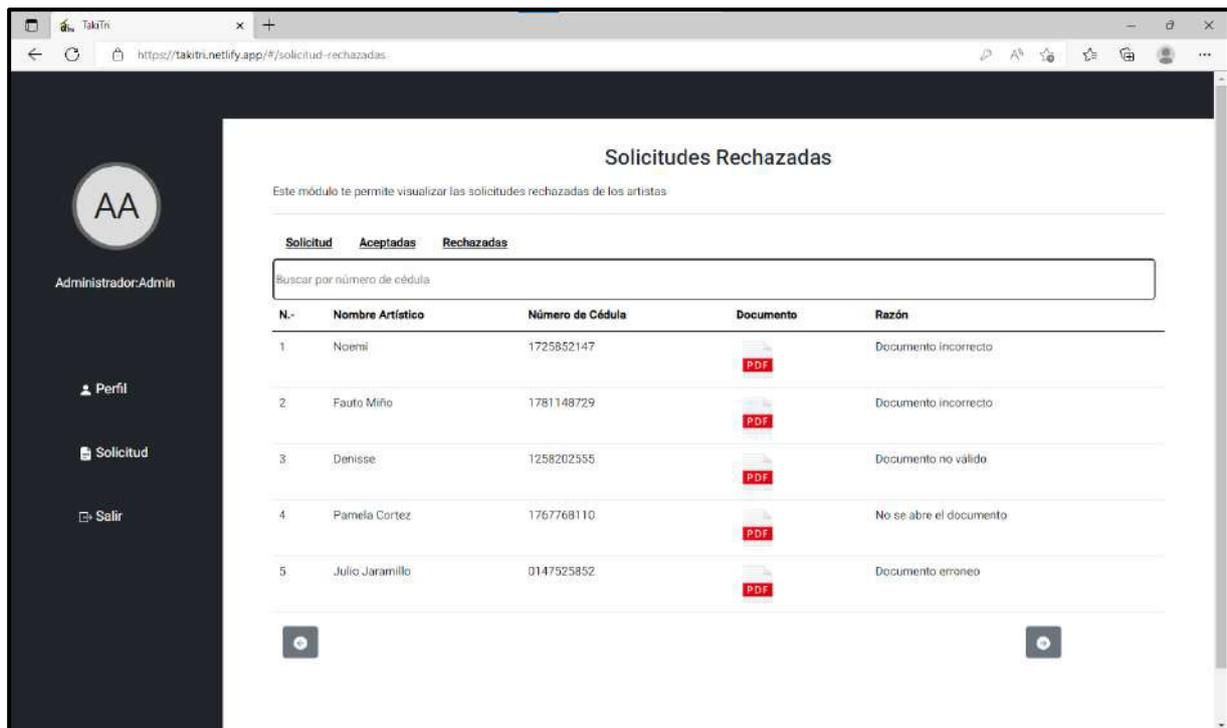


Fig.11: Solicitudes rechazadas.

3.3 Sprint 3. Resultado de las interfaces para el usuario artista.

Las tareas para el Sprint 3 está conformado por:

- Consumir varios métodos para registrar el rol artista.
- Consumir varios métodos para enviar solicitud de afiliación.
- Consumir varios métodos para gestionar los géneros musicales.
- Consumir varios métodos para gestionar álbumes.
- Consumir varios métodos para gestionar canciones.
- Buscar canciones

Consumir varios métodos para registrar el rol artista

En el *frontend* el usuario con perfil artista tiene la posibilidad de registrarse mediante un formulario en el cual debe ingresar la siguiente información: nombre, apellido, correo electrónico, contraseña, fecha de nacimiento y alias. Cabe recalcar que esta vista del registro está disponible para el usuario con perfil invitado. Todo esto es posible gracias al consumo de varios métodos por parte del *backend*, que es presentado en los componentes visuales del *frontend* como se puede ver en la **Fig.12**. Por otra parte, se detalla el funcionamiento completo que se puede ver de mejor manera en el **ANEXO III** del actual documento.

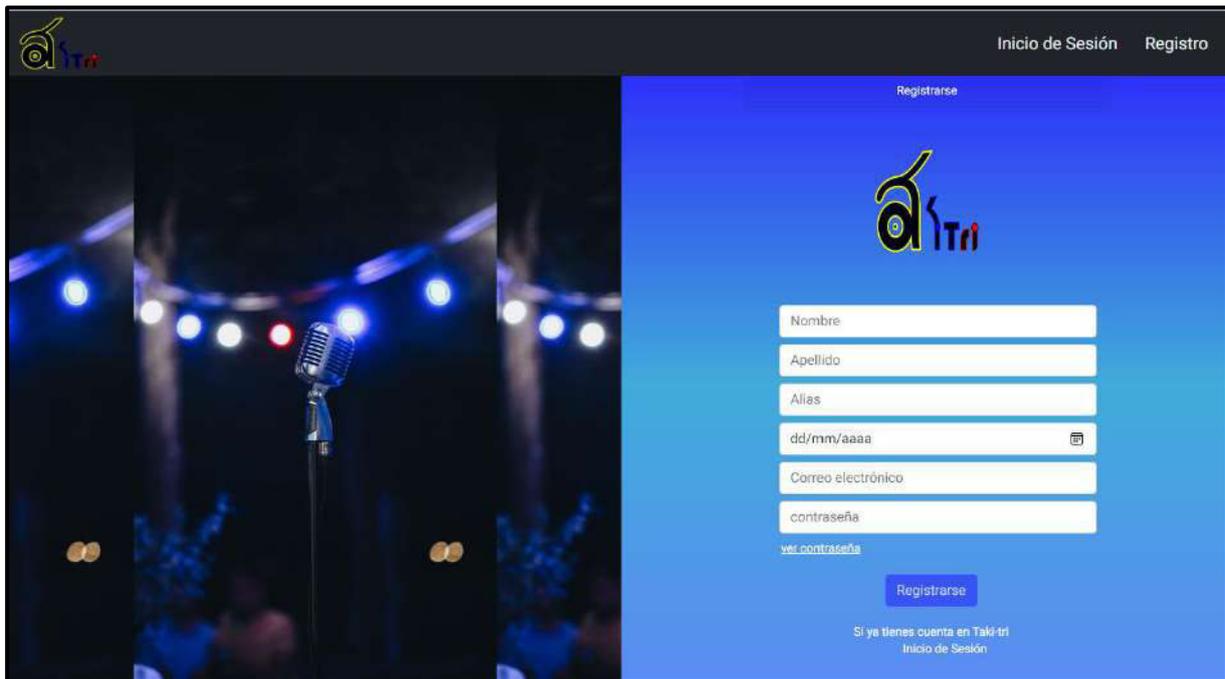


Fig.12: Registro del rol artista.

Consumir varios métodos para enviar solicitud de afiliación

En el *frontend* el usuario con perfil artista tiene la posibilidad de enviar una solicitud mediante un formulario llenando la siguiente información: nombre artístico, número de cédula y un documento en formato pdf. Además, esta actividad la puede realizar varias veces hasta que sea aprobada su solicitud por medio del administrador y que de esta manera el artista pueda tener acceso a más módulos dentro del *frontend*, caso contrario solo puede navegar bajo las páginas, como perfil y solicitud. Todo esto es posible gracias al consumo de varios métodos por parte del *backend*, que es presentado en los componentes visuales del *frontend* como se puede ver en la **Fig.13**. Por otra parte, se detalla el funcionamiento completo que se puede ver de mejor manera en el **ANEXO III** del actual documento.

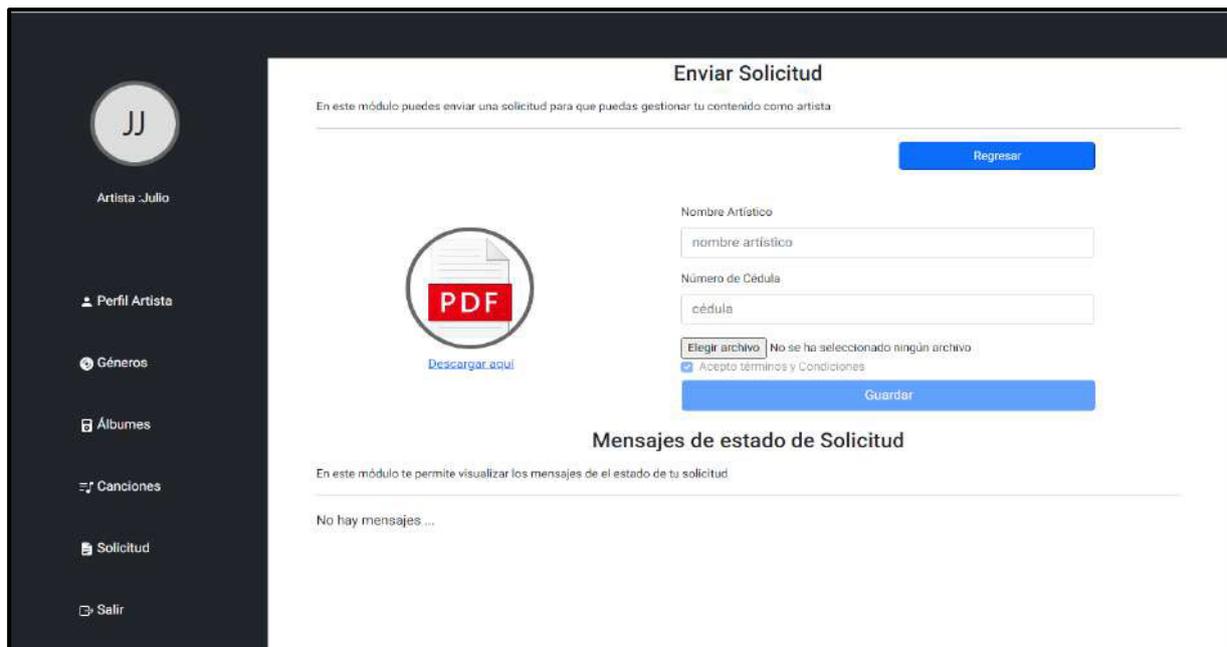


Fig.13: Enviar solicitud de afiliación.

Consumir varios métodos para gestionar los géneros musicales

En el *frontend* el usuario con perfil artista tiene la posibilidad de crear géneros musicales, mediante un formulario llenando la siguiente información: nombre del género, e imagen, también tiene la opción de eliminar géneros, esta actividad solo se la realiza cuando no contenga álbumes ni canciones en él mismo. Todo esto es posible gracias al consumo de varios métodos por parte del *backend*, que es presentado en los componentes visuales del *frontend* como se puede ver en las **Fig.14**, **Fig.15**. Por otra parte, se detalla el funcionamiento completo que se puede ver de mejor manera en el **ANEXO III** del actual documento.

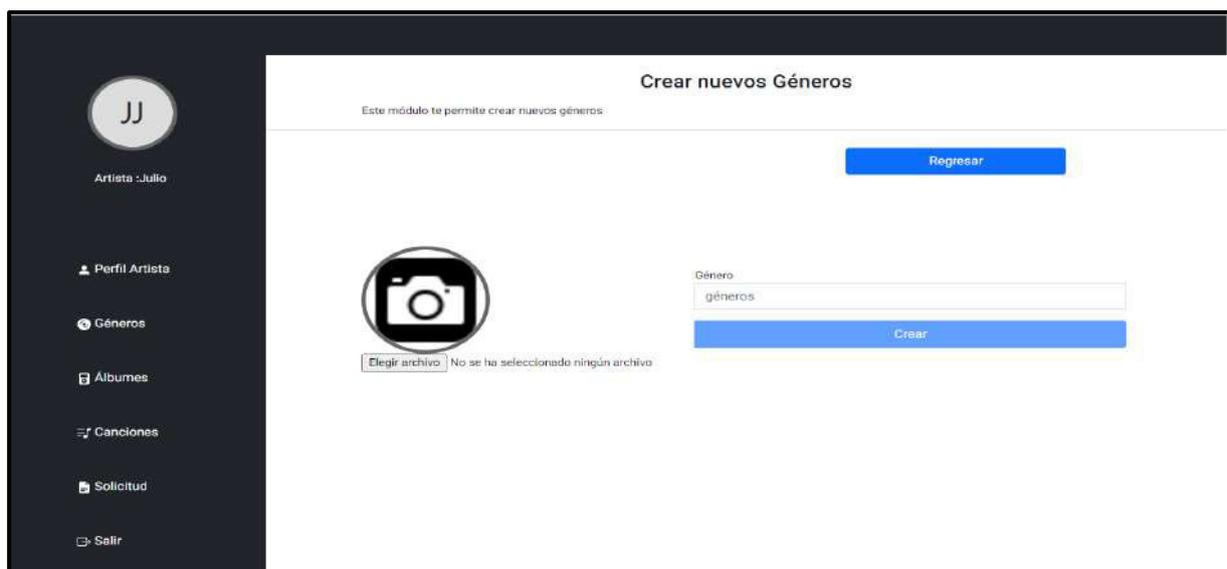


Fig.14: Crear género musical.

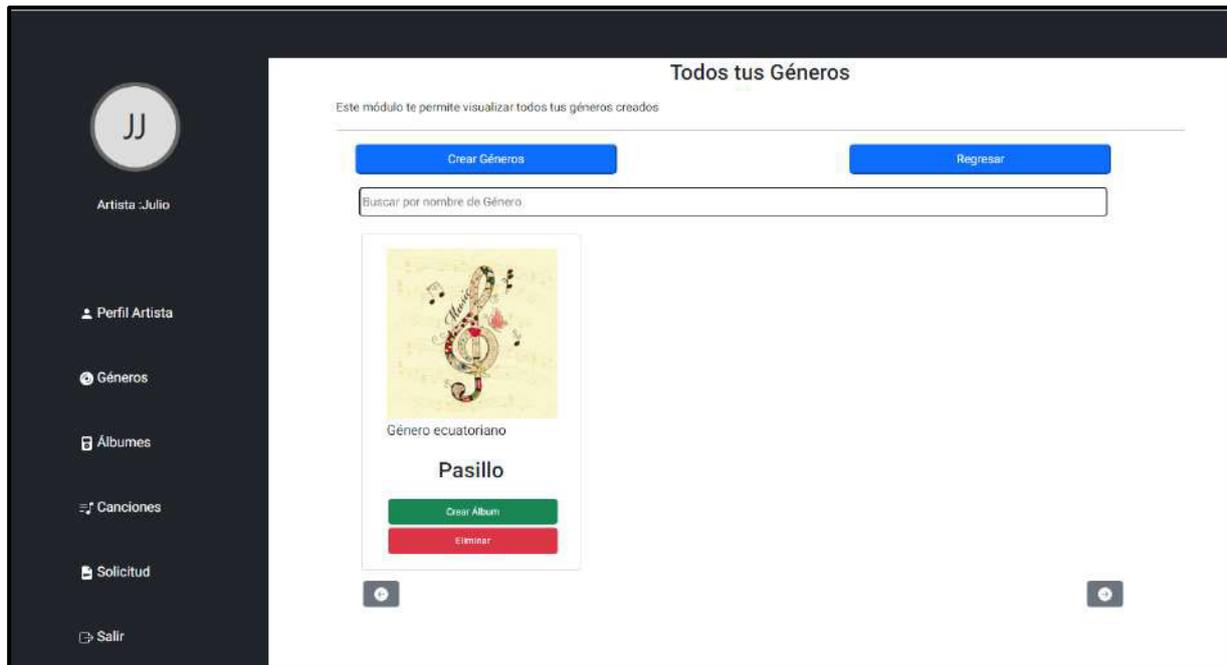


Fig.15: Gestión de Géneros.

Consumir varios métodos para gestionar álbumes

En el *frontend* el usuario con perfil artista tiene la posibilidad de crear álbumes, mediante un formulario llenando la siguiente información: nombre del álbum, nombre del autor, año e imagen, también tiene la opción de eliminar álbumes, esta actividad solo se la realiza cuando no contenga canciones en él. Es importante mencionar que para que el usuario artista se le permita crear un álbum debe tener un género creado con anterioridad. Todo esto es posible gracias al consumo de varios métodos por parte del *backend*, que es presentado en los componentes visuales del *frontend* como se puede ver en las **Fig.16**, **Fig.17** . Por otra parte, se detalla el funcionamiento completo que se puede ver de mejor manera en el **ANEXO III** del actual documento.

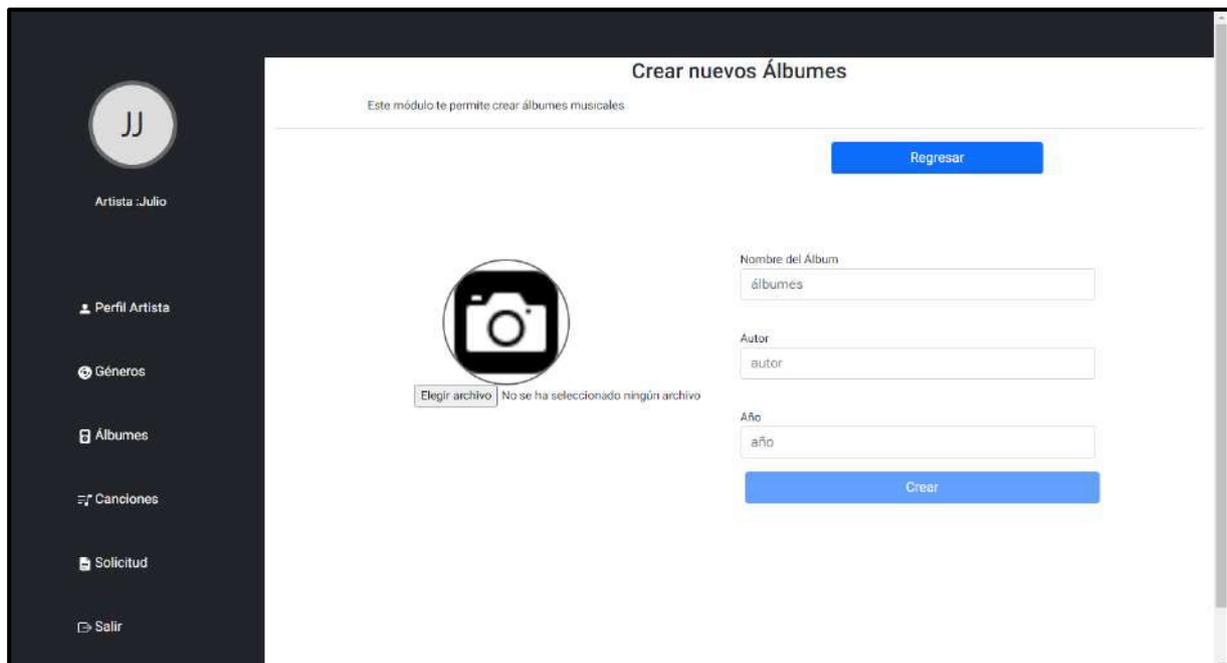


Fig.16 Crear álbum.

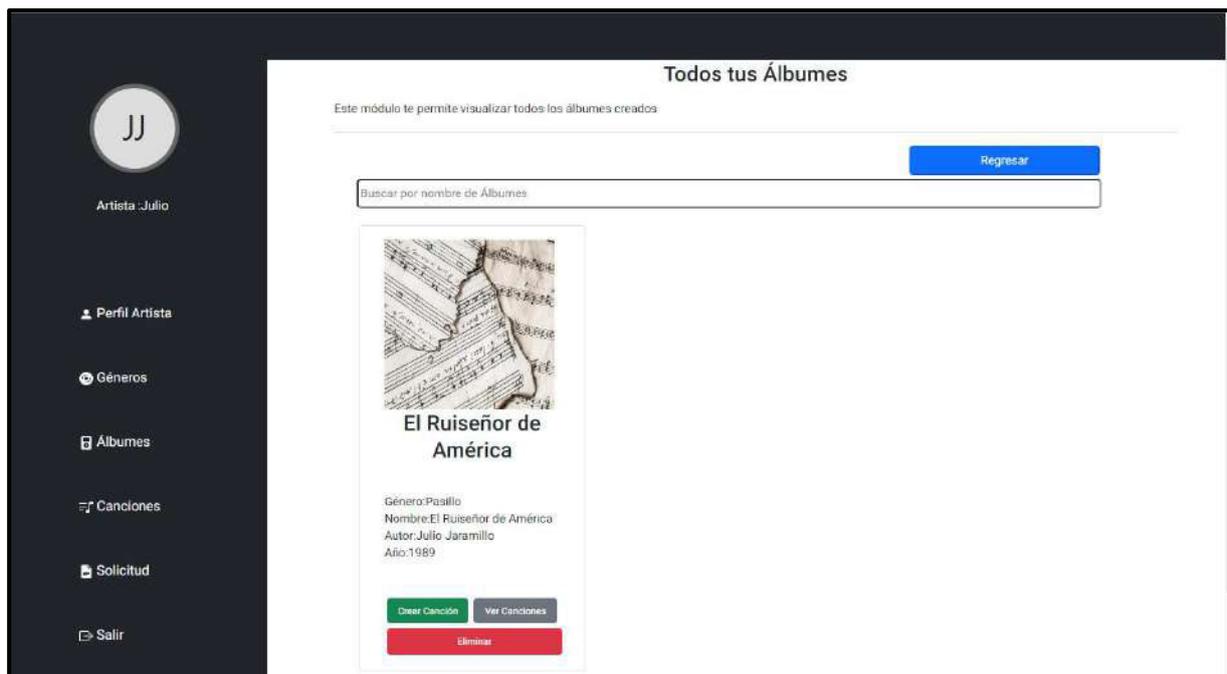


Fig.17: Gestión de Álbum.

Consumir varios métodos para gestionar canciones.

En el *frontend* el usuario con perfil artista tiene la posibilidad de crear canciones, mediante un formulario llenando la siguiente información: nombre de la canción y un archivo mp3, además de poder editar el mismo en los campos seleccionados que son: nombre de la canción y archivo mp3, también tiene la opción de eliminar canciones. Es importante mencionar que

para que el usuario artista se le permita crear canciones debe tener creado un género y álbum con anterioridad. Todo esto es posible gracias al consumo de varios métodos por parte del *backend*, que es presentado en los componentes visuales del *frontend* como se puede ver en las **Fig.18**, **Fig. 19**, **Fig. 20**. Por otra parte, se detalla el funcionamiento completo que se puede ver de mejor manera en el **ANEXO III** del actual documento.

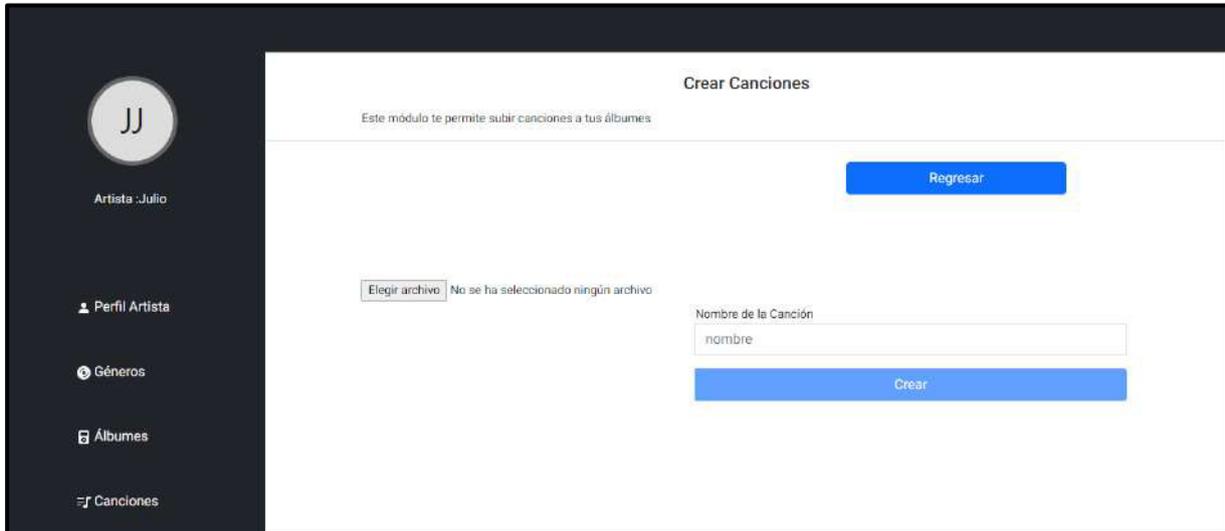


Fig.18: Subir canciones.

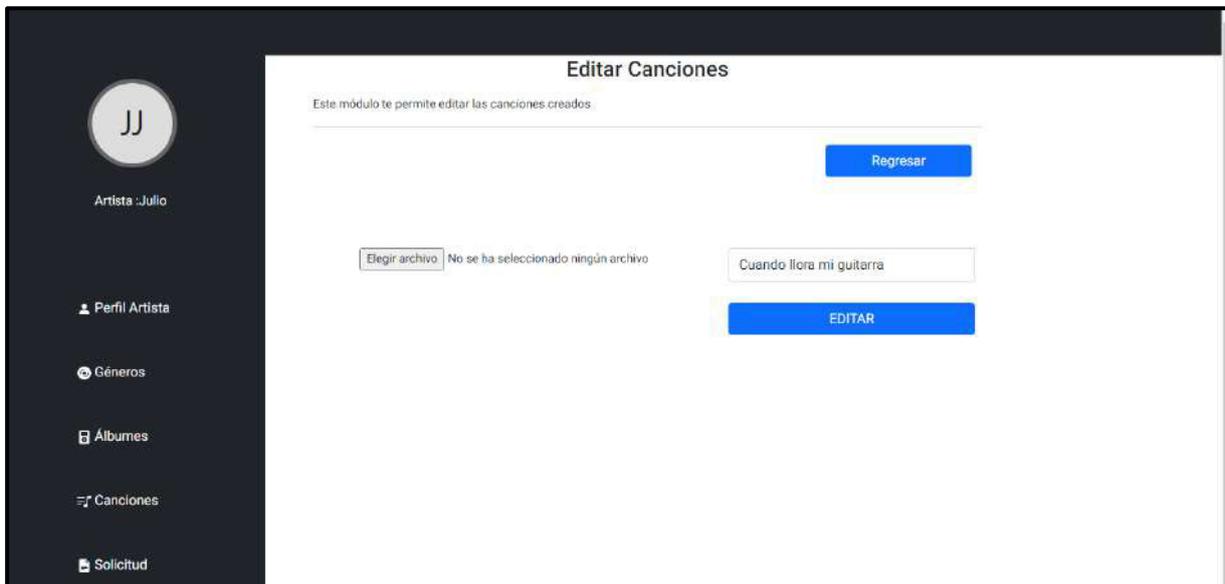


Fig. 19: Editar canciones.

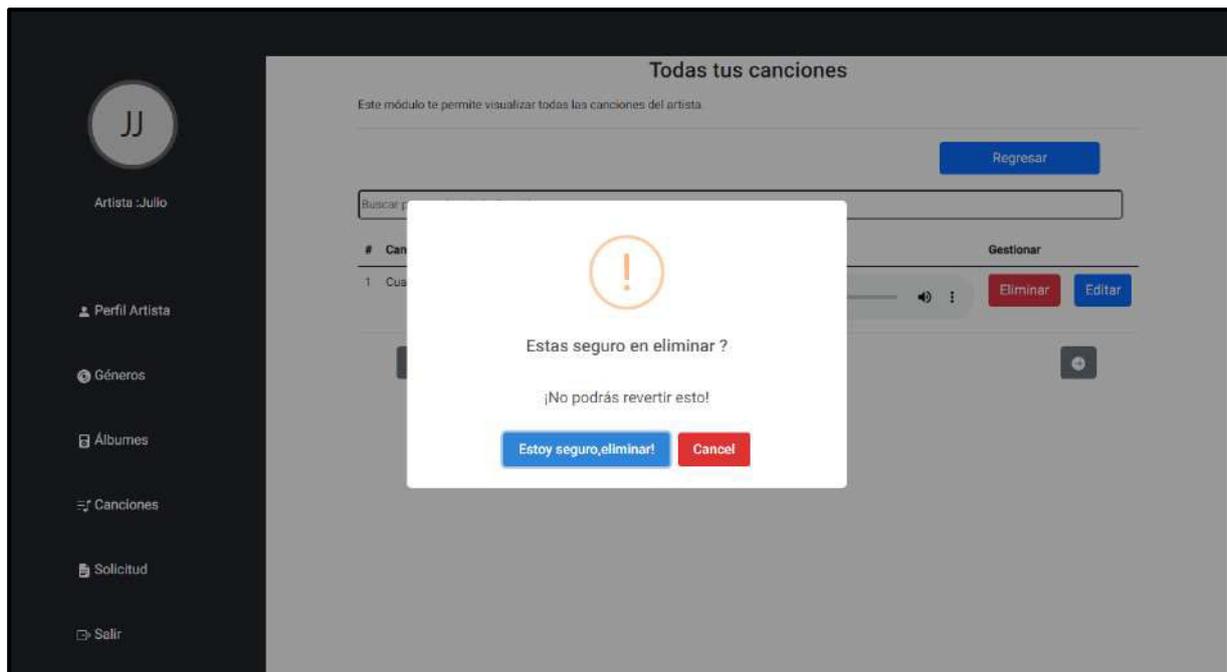


Fig. 20: Eliminar canciones.

3.4 *Sprint 4*. Resultado de las interfaces para el usuario invitado

La tarea para el Sprint 4 está conformado por:

Buscar canciones

En el *frontend* el usuario con perfil artista tiene la posibilidad de buscar canciones ubicando en el componente respectivo el nombre de la canción y así obtener los resultados correspondientes para luego poder reproducirla. Todo esto es posible gracias al desarrollo e implementación de un método propio por parte del *frontend* como se puede ver en la **Fig.21**. Por otra parte, se detalla el funcionamiento completo que se puede ver de mejor manera en el **ANEXO III** del actual documento.

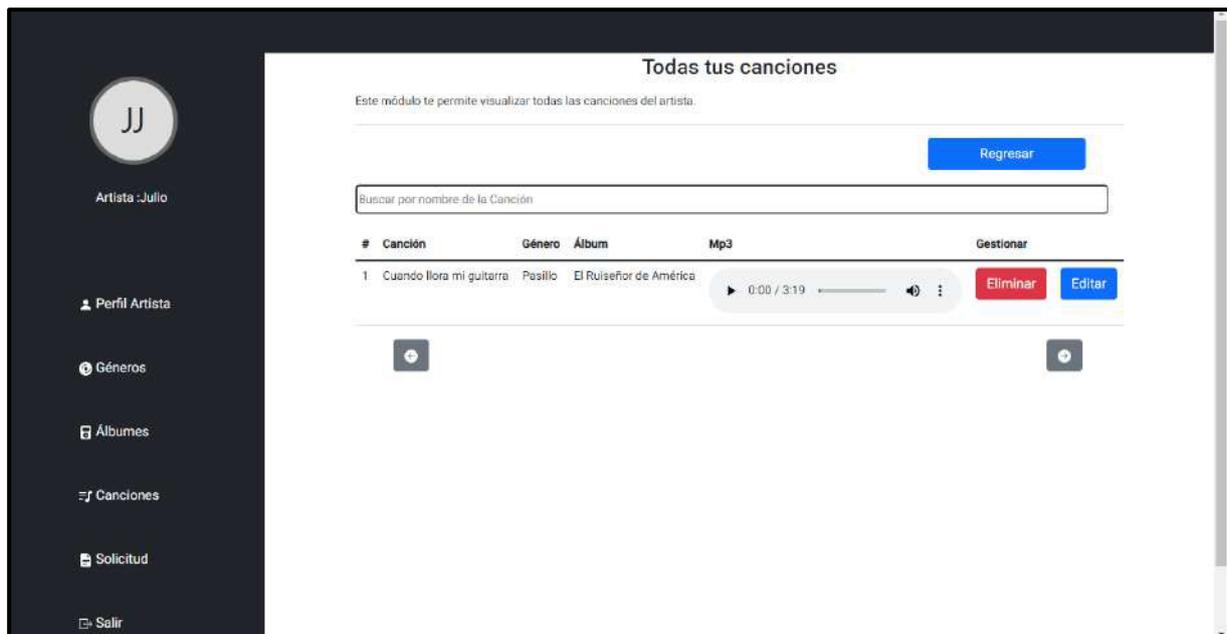


Fig.21: Buscar canciones.

3.5 Sprint 5 Pruebas y despliegue en el *frontend*

Se ha culminado de forma adecuada la etapa de codificación y ahora lo que se debe realizar es cumplir todas las tareas para el *Sprint 5*, las cuales están conformadas por:

- Aplicar y obtener resultados de pruebas unitarias.
- Aplicar y obtener resultados de pruebas de carga.
- Aplicar y obtener resultados de pruebas de compatibilidad.
- Aplicar y obtener resultados de pruebas de aceptación.
- Despliegue del *frontend*.

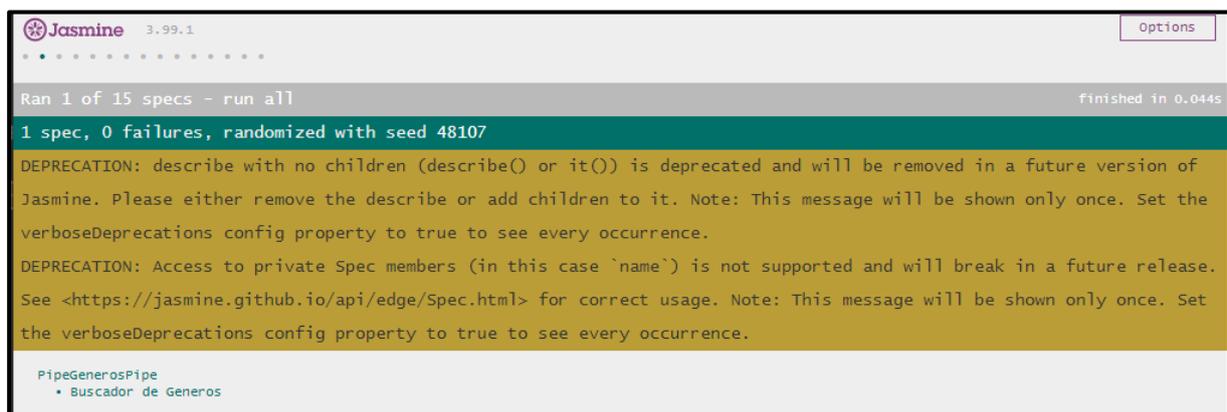
Aplicar y obtener resultados de pruebas unitarias

Para poder demostrar que una parte del código funciona de forma adecuadamente y de la manera prevista, se ha realizado las pruebas unitarias, dichas pruebas permiten testear y evaluar el código que se ha desarrollado, así como su funcionalidad es por esa razón, que este tipo de pruebas son de suma importancia en esta etapa de la metodología ya que de esta manera se puede obtener un producto *software* de calidad. Esas pruebas funcionan de la siguiente manera, se toma una parte del código de un módulo para luego probar con la herramienta que se ha seleccionado y posterior a ello visualizar los resultados que se han obtenido y corregir los errores que se puedan presentar [31]. Por otra parte, la herramienta que se ha seleccionado para este tipo de prueba es Karma [32].

En la **Fig.22**, se visualiza una fracción de código que se ha implementado en una prueba unitaria, por otra parte, en la **Fig. 23**, se muestra el resultado que se ha obtenido, mientras que el detalle completo de la ejecución de esta prueba y los resultados que se han obtenido se pueden ver de mejor manera en el **ANEXO II** del actual documento.

```
1 import { PipeGenerosPipe } from './pipe-generos.pipe';
2
3 describe('PipeGenerosPipe', () => {
4   it('Buscador de Generos', () => {
5     const pipe = new PipeGenerosPipe();
6     expect(pipe).toBeTruthy();
7   });
8 });
9
```

Fig.22: Fracción de código de la prueba unitaria.



The screenshot shows the Jasmine test runner interface. At the top, it displays 'Jasmine 3.99.1' and an 'Options' button. Below that, it indicates 'Ran 1 of 15 specs - run all' and 'finished in 0.044s'. A green bar shows '1 spec, 0 failures, randomized with seed 48107'. Two deprecation warnings are shown in yellow: 'DEPRECATION: describe with no children (describe() or it()) is deprecated...' and 'DEPRECATION: Access to private Spec members (in this case `name`) is not supported...'. At the bottom, a tree view shows 'PipeGenerosPipe' with a sub-item 'Buscador de Generos'.

Fig. 23: Resultado de prueba unitaria.

Una vez que se ha finalizado las pruebas unitarias y con los resultados de la herramienta que se ha seleccionado se determina que todos los módulos del *frontend* no presenta ningún fallo en el código o funcionalidad pertinente.

Aplicar y obtener resultados de pruebas de carga

Las pruebas de carga dentro del desarrollo del *frontend* se encargan de verificar mediante varios simuladores la interacción de varios usuarios con funcionalidades específicas de forma simultánea para poder visualizar el rendimiento de la misma [33]. Es por esta razón, que para realizar esta prueba se utiliza la aplicación web *PageSpeed Insights* de *Google*, la cual permite ejecutar pruebas de carga y obtener en tiempo real mediante porcentajes el nivel de rendimiento que soporta el *frontend*, así como optimizar los tiempos de carga. En ese sentido, en la **Fig.24**, se visualiza el resultado de la ejecución de la respectiva prueba.



Fig.24: Frontend bajo una prueba de carga.

Una vez que se ha finalizado la prueba de carga y con los resultados de la aplicación que se ha seleccionado se determina que todos los módulos del *frontend* están bien optimizados y que el rendimiento de carga es totalmente aceptable para ser puesto a producción

Aplicar y obtener resultados de pruebas de compatibilidad

Las pruebas de compatibilidad se describen como una verificación a nivel de interfaces de usuario y presentación de información de un producto *software*. La idea de este tipo de pruebas es verificar que exista simetría en cada una de las interfaces al momento de ser visualizada en diferentes dispositivos electrónicos y navegadores web [34]. Es por esta razón que se procede a realizar esta prueba en distintos navegadores como se observa en la **TABLA VIII**. Por otra parte, el detalle completo de la ejecución de esta prueba y los resultados que se han obtenido se pueden ver de mejor manera en el **ANEXO II** del actual documento.

TABLA VIII: Navegadores que intervienen en las pruebas.

NOMBRE	VERSIÓN	OBSERVACIÓN
Google Chrome	103.0.5060.134	Completamente funcional
Microsoft Edge	103.0.1264.71	Completamente funcional
Opera	89.0.4447.71	Completamente funcional

Una vez que se ha finalizado las pruebas de aceptación y con los resultados en diferentes navegadores se determina que el *frontend* y su contenido se adapta correctamente, sin fallas en la presentación de información y funcionamiento de módulos.

Aplicar y obtener resultados de pruebas de aceptación

Este tipo de pruebas permiten determinar si el *software* cumple con todas las expectativas que el usuario final ha solicitado y certifique que el mismo se ha realizado bajo los requerimientos ya determinados [35]. Es por esta razón que en la **TABLA IX** se visualiza la primera prueba de aceptación que se ha realizado bajo los requerimientos que el usuario ha mencionado. Por otra parte, el detalle completo de la ejecución de esta prueba y los resultados que se han obtenido se pueden ver de mejor manera en el **ANEXO II** del actual documento.

TABLA IX: Pruebas de aceptación 1 - Visualizar página informativa.

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA001	Identificador de historia de Usuario: HU001
Nombre: Visualizar una página informativa	
Descripción: El usuario invitado en el <i>frontend</i> pueden visualizar la página informativa, además de interactuar en el menú principal con la siguiente información: <ul style="list-style-type: none">• Misión y Visión• Programadora.• Inicio	
Pasos de ejecución: Para iniciar sesión usuario artista y administrador: <ul style="list-style-type: none">• Ubicar la URL del <i>frontend</i> en los navegadores.• Hacer clic en el menú superior en las diferentes opciones como lo son: programadora, misión y visón.	
Resultado deseado: En TakiTri del <i>frontend</i> admite visualizar una página informativa.	
Evaluación de la prueba: Se evidencia el resultado esperado. Conformidad del cliente 100%.	

Una vez que se ha finalizado las pruebas de aceptación y en conjunto con la aceptación del *Product Owner* se determina que se cumple con todos los requerimientos que se han estipulado al inicio del proyecto.

Despliegue del *frontend*

Una vez que se ha finalizado la etapa de pruebas, el siguiente paso es realizar el despliegue a producción como parte del ciclo de desarrollo de un producto *software*. En ese sentido, la herramienta que ha permitido el despliegue del *frontend* es *Netlify*, el cual proporciona un servicio de *hosting* totalmente gratuito en donde se encuentra alojada la aplicación TakiTri y que la misma pueda ser accedida desde cualquier navegador web. No obstante, este procedimiento se encuentra detallado de mejor manera en el **ANEXO IV** del presente documento. Sin embargo, en la **Fig. 25**, se muestra el resultado final de haber realizado el despliegue del *frontend* a producción bajo un dominio, la cual se puede acceder digitando la siguiente url en cualquier navegador web.

<https://takitri.netlify.app>

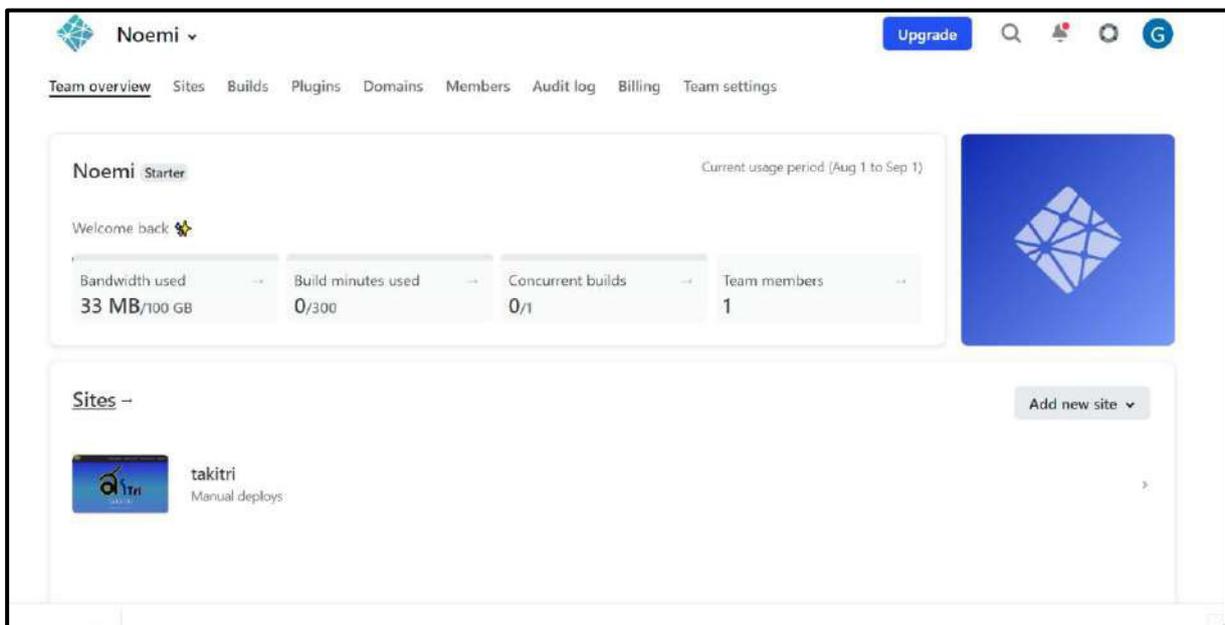


Fig. 25: Despliegue del *frontend*.

4 CONCLUSIONES

En esta sección del presente documento se visualiza las conclusiones que se han logrado en el proceso de desarrollo del *frontend* del actual proyecto de Integración Curricular.

- Se concluye que el proyecto denominado “TakiTri” cumple con el alcance y cada uno de los objetivos que se han establecido, logrando de esta manera que los artistas ecuatorianos a través de una aplicación del lado del cliente (*frontend*) puedan gestionar sus canciones y tener un acercamiento significativo con el pueblo ecuatoriano de una forma gratuita e interactiva.
- La aplicación y ejecución de la metodología *Scrum* en el actual trabajo de integración curricular ha permitido que el desarrollo del *frontend* se lo pueda culminar en el tiempo que se ha establecido al igual que las actividades de cada *Sprint*.
- Establecer los requerimientos como uno de los puntos de partida en el desarrollo del *frontend* ha permitido seleccionar de forma correcta los artefactos y roles en la metodología ágil, así como herramientas para la codificación, pruebas y despliegue.
- Establecer los diseños como uno de los puntos iniciales del actual trabajo de integración curricular ha permitido poder tomar en cuenta la experiencia de usuario desde los inicios del *frontend*.
- Al implementar el MVC, se ha logrado acceder a la información de la base de datos para poder visualizarlos, editarlos y eliminarlos en el *frontend*.
- La correcta implementación de herramientas y librerías para la codificación del *frontend* ha sido de gran aporte ya que cada una de ellas tiene una alta compatibilidad e integración entre las otras librerías del componente logrando que el desarrollo sea mucho más fácil y ordenado.
- Al establecer una conexión estable con el *backend*, se ha verificado que TakiTri puede enviar información y recibirla de forma segura en tiempo real.
- Las diversas pruebas que se ha implementado en el *frontend*, dan como resultado un adecuado funcionamiento a nivel de los módulos, presentación de información, adaptabilidad y rendimiento.

5 RECOMENDACIONES

En esta sección del presente documento se visualiza las recomendaciones que se han logrado en el proceso de desarrollo del *frontend* del actual proyecto de Integración Curricular.

- Es recomendable que se sigan implementando más funcionalidades para poder satisfacer aún más las necesidades del usuario.
- Es recomendable capacitar a los administradores y artistas para que puedan gestionar su contenido de una forma apropiada.
- Es recomendable que se implemente más mecanismos de seguridad a los datos que sean consumidos por parte del *frontend*, ya que día a día existen más formas de hacer vulnerable un sistema *software*.
- Es recomendable que se controle el ingreso de los datos de una forma más sigilosa.
- Es importante que se implemente más seguridades para los artistas al momento de subir sus canciones a TakiTri.
- Es recomendable mencionar a los artistas que sus datos obtenidos están seguros en el *frontend* y que su información recopilada es de uso exclusivo para el componente.

6 REFERENCIAS BIBLIOGRÁFICAS

- [1] Instituto Nacional de Patrimonio Cultural, “El Pasillo ecuatoriano – Instituto Nacional de Patrimonio Cultural.” <https://www.patrimoniocultural.gob.ec/el-pasillo-ecuadoriano/> (accessed May 31, 2022).
- [2] ElUniverso, “La danza y el sanjuanito también tienen festejos | Intercultural | Noticias | El Universo.” <https://www.eluniverso.com/vida-estilo/2015/06/23/nota/4979551/danza-sanjuanito-tambien-tienen-festejos/> (accessed May 31, 2022).
- [3] Spotify, “Escuchar es todo - Spotify.” <https://www.spotify.com/ec/> (accessed May 31, 2022).
- [4] Youtube, “YouTube.” <https://www.youtube.com/> (accessed May 31, 2022).
- [5] Carlos Bolaños, “Spotify: el reggaetón fue lo más escuchado en Ecuador en 2020,” Dec. 02, 2020. <https://www.metroecuador.com.ec/ec/noticias/2020/12/02/spotify-reggaeton-fue-lo-mas-escuchado-ecuador-2020-aqui-los-artistas-canciones-mas-reproducidas.html> (accessed May 31, 2022).
- [6] Servicio Nacional de Derechos Intelectuales, “Derechos Conexos – Servicios.” <https://www.derechosintelectuales.gob.ec/derechos-conexos/> (accessed May 31, 2022).
- [7] S. Graciela, P. Ibarra, R. Quispe, F. F. Mullicundo, D. A. Lamas, and L. Presente, “HERRAMIENTAS Y TECNOLOGÍAS PARA EL DESARROLLO WEB DESDE EL FRONTEND AL BACKEND”, Accessed: May 31, 2022. [Online]. Available: <https://www.campusmvp.es/recursos/post/Desar>
- [8] E. Serna, R. Martínez, and P. Tamayo, “Una revisión a la realidad de la automatización de las pruebas del software A Review of Reality of Software Test Automation,” vol. 23, no. 1, pp. 169–183, 2019, doi: 10.13053/CyS-23-1-2782.
- [9] Jhonny Caicedo, “La Industria Musical en las Plataformas Digitales en el Año 2021: ¿Cómo Emprender con la Música en la Era Digital?”.
- [10] Angular, “Angular.” <https://angular.io/> (accessed May 31, 2022).
- [11] Vue.js, “Vue.js - The Progressive JavaScript Framework.” <https://vuejs.org/> (accessed May 31, 2022).
- [12] J. Garcia, B. J. Plat, and P. Salazar, “Metodologías Ágiles en el Desarrollo de Software”, Accessed: May 31, 2022. [Online]. Available: www.agileuniverse.com.
- [13] Zuñiga Vazquez and Luis Ramon, “Desarrollo de aplicaciones web utilizando Angular como framework”.
- [14] M. T. Nevado and M. R. Palero, “Estudio de metodologías ágiles. Aplicación SCRUM”.
- [15] “> ¿Qué es figma? La mejor herramienta de prototipado web.” <https://cei.es/que-es-figma/> (accessed Jun. 01, 2022).
- [16] J. Camilo, G. Mejía, F. A. Vargas Agudelo, and K. G. Gil, “Marco de Trabajo para Seleccionar un Patrón Arquitectónico en el Desarrollo de Software”.

- [17] Demián Gutierrez, "Métodos de Desarrollo de Software." <https://silo.tips/download/metodos-de-desarrollo-de-software> (accessed Jun. 02, 2022).
- [18] "Cloud Storage para Firebase Documentation." <https://firebase.google.com/docs/storage?hl=es-419> (accessed Jun. 01, 2022).
- [19] Víctor Alberto Iranzo Jiménez, "Desarrollo de software basado en microservicios." <https://riunet.upv.es/bitstream/handle/10251/111173/Iranzo%20-%20Desarrollo%20de%20software%20basado%20en%20microservicios%3a%20un%20caso%20de%20estudio%20para%20evaluar%20sus%20ven....pdf?sequence=1&Allowed=y> (accessed Jun. 01, 2022).
- [20] C. Blanco Bueno, "Ingeniería del Software II."
- [21] M. D. Panizzi, G. Hodes, F. Ortiz, P. Vazquez, and R. A. Bertone, *Despliegue de sistemas de software*. [Online]. Available: <https://www.cessi.org.ar/opssi>.
- [22] U. Autónoma De Madrid *et al.*, "Métodos de investigación educativa [EL ESTUDIO DE CASOS]."
- [23] Jesús Santiago Rial Huerta, "Aplicación de Metodologías Ágiles a Desarrollo de Proyectos".
- [24] E. Bahit, "Scrum y eXtreme Programming para Programadores."
- [25] J. L. Comesaña, "DISEÑO DE INTERFACES WEB Desarrollo de Aplicaciones Web."
- [26] C. Reynoso, "Arquitectura de Sowane Manual completo."
- [27] Edison Patricio Farinango Caiza, "ESTUDIO DEL FRAMEWORK ANGULAR PARA DESARROLLAR APLICACIONES SINGLE-PAGE. DESARROLLO DEL SISTEMA DE INFORMACIÓN, SEGUIMIENTO Y CONTROL PARA LA HACIENDA GANADERA".
- [28] José Carlos Fatjó, "Introducción a Angular Material." <https://tech.tribalyte.eu/blog-introduccion-angular-material> (accessed Jun. 02, 2022).
- [29] Arunkumar Gudelli, "Angular Material Icons." <https://www.angularjswiki.com/angular/angular-material-icons-list-mat-icon-list/> (accessed Jun. 02, 2022).
- [30] parzibyte, "Sweet Alert 2 - Tutorial con ejemplos." <https://parzibyte.me/blog/2019/12/16/sweet-alert-2-tutorial-ejemplos/> (accessed Jun. 02, 2022).
- [31] YeePLY, "¿Qué son las pruebas unitarias y cómo llevar una a cabo?" <https://www.yeePLY.com/blog/que-son-pruebas-unitarias/> (accessed Jun. 13, 2022).
- [32] Claudio Coello, "Cómo usar Testing en Angular con Jasmine y Karma." <https://digital55.com/como-usar-testing-angular-jasmine-karma/> (accessed Jun. 13, 2022).
- [33] Carlos Mario Zapata J.1 Ph.D. y Christian de Jesús Cardona Velásquez. M, "Comparación de las características de algunas herramientas de software para pruebas de carga." <https://www.redalyc.org/pdf/1331/133119867014.pdf> (accessed Jun. 13, 2022).

- [34] Unguess, "Pruebas de compatibilidad de dispositivos: qué son y cómo elegir los dispositivos adecuados." <https://blog.unguess.io/es/pruebas-de-compatibilidad-de-dispositivos-que-son-y-como-elegir-los-dispositivos-adecuados> (accessed Jun. 13, 2022).
- [35] Digité, "Pruebas De Aceptación: El Qué Y Porqué Los Tipos Que Hay Que Conocer." <https://www.digite.com/es/agile/pruebas-de-aceptacion/> (accessed Jun. 13, 2022).

7 ANEXOS

A continuación, se presenta cada uno de los Anexos que se ha utilizado para el desarrollo del *frontend*, los cuales se encuentran detallados de la siguiente manera:

- **ANEXO I.** Resultado del programa antiplagio Turnitin.
- **ANEXO I.** Manual Técnico.
- **ANEXO II.** Manual de Usuario.
- **ANEXO III.** Manual de Instalación.

ANEXO I

A continuación, se presenta el certificado que el Director de Tesis ha emitido y en donde se evidencia el resultado que se ha obtenido en la herramienta antiplagio Turnitin.



**ESCUELA POLITÉCNICA NACIONAL
ESCUELA DE FORMACIÓN DE TECNÓLOGOS
CAMPUS POLITÉCNICO "ING. JOSÉ RUBÉN ORELLANA"**

CERTIFICADO DE ORIGINALIDAD

Quito, D.M. 22 de agosto de 2022

De mi consideración:

Yo, Loarte Cajamarca Byron Gustavo, en calidad de Director del Trabajo de Integración Curricular titulado Desarrollo de un frontend asociado al DESARROLLO DE SISTEMA WEB Y APLICACIÓN MÓVIL INFORMATIVA SOBRE MÚSICA NACIONAL ECUATORIANA elaborado por la estudiante Denisse Noemi Cumbal Ganchozo de la carrera en Tecnología Superior en Desarrollo de Software, certifico que he empleado la herramienta Turnitin para la revisión de originalidad del documento escrito secciones: Descripción del componente desarrollado, Metodología, Resultados, Conclusiones y Recomendaciones, producto del Trabajo de Integración Curricular indicado.

El documento escrito tiene un índice de similitud del 11%.

Es todo cuanto puedo certificar en honor a la verdad, pudiendo el interesado hacer uso del presente documento para los trámites de titulación.

NOTA: Se adjunta el informe generado por la herramienta Turnitin.

Atentamente,

**Loarte Cajamarca Byron Gustavo
Profesor Ocasional a Tiempo Completo
Escuela de Formación de Tecnólogos**

ANEXO II

Requerimientos que se han recopilado

En la **Tabla X** se visualiza los requerimientos alcanzados al inicio del proyecto en donde concuerda con lo solicitado por el usuario.

Tabla X:Requerimientos recopilados.

RECOPIACIÓN DE REQUERIMIENTOS		
TIPO DEL SISTEMA	ID - RR	ENUNCIADO DEL ÍTEM
FRONTEND	RR002	El usuario con perfil administrador y artista necesitan consumir varios métodos para: <ul style="list-style-type: none"> • Iniciar sesión • Cerrar sesión. • Recuperar contraseña.
	RR003	El usuario con perfil administrador necesita consumir varios métodos para <ul style="list-style-type: none"> • Modificar perfil de usuario.
	RR004	El usuario con perfil administrador necesita consumir varios métodos para: <ul style="list-style-type: none"> • Gestionar solicitudes de artistas.
	RR005	El usuario con perfil invitado necesita consumir un método para: <ul style="list-style-type: none"> • Registrarse a través de un formulario.
	RR006	El usuario con perfil artista necesita consumir varios métodos para <ul style="list-style-type: none"> • Modificar perfil de usuario. • Enviar solicitud
	RR007	El usuario con perfil artista necesita consumir varios métodos para: <ul style="list-style-type: none"> • Gestionar géneros musicales.
	RR008	El usuario con perfil artista necesita consumir varios métodos para: <ul style="list-style-type: none"> • Gestionar álbumes.
	RR009	El usuario con perfil artista necesita consumir varios métodos para:
	RR009	El usuario con perfil artista necesita consumir varios métodos para:

		<ul style="list-style-type: none"> • Gestionar canciones.
	RR010	El usuario con perfil artista necesita realizar lo siguiente: <ul style="list-style-type: none"> • Buscador de canciones.

Historias de Usuario

Terminada la etapa de recolección de requerimientos, se da paso a crear las Historias de usuarios para el *frontend*. Es por ello, que se presentan las 9 Historias de usuario guiadas en base a los requerimientos que comienzan desde **TABLA XI** hasta **TABLA XIX**.

TABLA XI: Ingreso y salida de TAKITRI por parte de los usuarios administrador y artista.

HISTORIA DE USUARIO	
Identificador: HU002	Usuario: Administrador & Artista
Nombre historia: Iniciar sesión, cerrar sesión y recuperar contraseña.	
Prioridad en Negocio: Alta	Riesgo en Negocio: Alta
Iteración asignada: 1	
Responsable: Denisse Cumbal	
Descripción: El usuario administrador y artista en el <i>frontend</i> puede consumir varios métodos para: <ul style="list-style-type: none"> • Iniciar sesión • Cerrar sesión. • Recuperar contraseña. 	
Observación: El usuario artista necesita estar registrado para acceder a los métodos anteriormente mencionados. Además, para que el usuario administrador y artista puedan recuperar su contraseña se envía un correo a su e-mail respectivamente.	

TABLA XII: Modificar perfil rol administrador.

HISTORIA DE USUARIO	
Identificador: HU003	Usuario: Administrador
Nombre historia: Modificar perfil de usuario.	
Prioridad en Negocio: Media	Riesgo en Negocio: Media
Iteración asignada: 1	
Responsable: Denisse Cumbal	
<p>Descripción: El usuario administrador en el <i>frontend</i> puede consumir varios métodos para visualizar y editar su perfil por medio de los siguientes campos:</p> <ul style="list-style-type: none"> • Nombre. • Imagen. • Entre otros. 	
<p>Observación:</p> <p>El usuario solo puede modificar su perfil como administrador siempre y cuando inicie sesión.</p>	

TABLA XIII: Gestionar solicitudes de artistas.

HISTORIA DE USUARIO	
Identificador: HU004	Usuario: Administrador
Nombre historia: Gestionar solicitudes de artistas	
Prioridad en Negocio: Alta	Riesgo en Negocio: Alta
Iteración asignada: 2	
Responsable: Denisse Cumbal	
<p>Descripción: El usuario administrador en el <i>frontend</i> puede consumir varios métodos para:</p> <ul style="list-style-type: none"> • Visualizar solicitudes de artistas. • Aprobar solicitudes de artistas. • Rechazar solicitudes de artistas. 	

Observación:

EL usuario administrador puede visualizar las solicitudes de los artistas para aprobarlas o rechazarlas.

Se rechazan las solicitudes de los artistas cuando no esté el documento correctamente enviado por el artista.

Se aprueban las solicitudes del artista cuando conste el documento enviado correctamente por el artista.

TABLA XIV: Registro del formulario.

HISTORIA DE USUARIO	
Identificador: HU005	Usuario: Invitado
Nombre historia: Registrarse a través de un formulario.	
Prioridad en Negocio: Media	Riesgo en Negocio: Media
Iteración asignada: 2	
Responsable: Denisse Cumbal	
Descripción: <p>El usuario artista en el <i>frontend</i> puede consumir varios métodos para registrarse por medio llenando los siguientes campos:</p> <ul style="list-style-type: none"> • Nombre. • Alias. • Fecha de nacimiento. • Correo electrónico. 	
Observación: <p>El usuario Invitado, debe llenar correctamente todos los campos solicitados en el registro, para ingresar a los diferentes módulos permitidos para el usuario artista</p>	

TABLA XV: Modificar solicitud artista y enviar solicitud.

HISTORIA DE USUARIO	
Identificador: HU006	Usuario: Artista
Nombre historia: Modificar perfil de usuario y enviar solicitud.	
Prioridad en Negocio: Media	Riesgo en Negocio: Media
Iteración asignada: 2	
Responsable: Denisse Cumbal	
<p>Descripción: El usuario artista en el <i>frontend</i> puede consumir varios métodos para modificar, visualizar su perfil además de enviar solicitudes por medio de los siguientes campos:</p> <p>Perfil</p> <ul style="list-style-type: none"> • Nombre. • Imagen de perfil. • Entre otros. <p>Solicitud</p> <ul style="list-style-type: none"> • Nombre artístico. • Número de cédula • Documento PDF. 	
<p>Observación:</p> <p>El usuario artista es el único que puede editar su perfil, siempre y cuando inicie sesión previamente, además el usuario artista puede enviar solicitudes para poder acceder a módulos para gestionar géneros, álbumes y canciones.</p>	

TABLA XVI: Gestionar géneros musicales.

HISTORIA DE USUARIO	
Identificador: HU007	Usuario: Artista
Nombre historia: Gestionar géneros musicales.	
Prioridad en Negocio: Media	Riesgo en Negocio: Media

Iteración asignada: 3
Responsable: Denisse Cumbal
<p>Descripción: El usuario artista en el <i>frontend</i> puede consumir varios métodos para:</p> <ul style="list-style-type: none"> • Crear géneros musicales. • Visualizar géneros musicales. • Eliminar géneros musicales.
<p>Observación:</p> <p>El usuario artista solo puede ver sus géneros registrados por el mismo, siempre y cuando inicie sesión previamente y halla enviado una solicitud previamente aceptada.</p>

TABLA XVII: Gestionar álbumes.

HISTORIA DE USUARIO	
Identificador: HU008	Usuario: Artista
Nombre historia: Gestionar álbumes.	
Prioridad en Negocio: Media	Riesgo en Negocio: Media
Iteración asignada: 3	
Responsable: Denisse Cumbal	
<p>Descripción: El usuario artista en el <i>frontend</i> puede consumir varios métodos para:</p> <ul style="list-style-type: none"> • Crear álbum. • Visualizar álbum. • Eliminar álbum. 	
<p>Observación:</p> <p>Para crear un álbum el artista debe seleccionar un género musical y haber enviado una solicitud previamente aceptada.</p>	

TABLA XVIII: Subir canciones.

HISTORIA DE USUARIO	
Identificador: HU009	Usuario: Artista
Nombre historia: Gestionar canciones.	
Prioridad en Negocio: Alta	Riesgo en Negocio: Alta
Iteración asignada: 3	
Responsable: Denisse Cumbal	
<p>Descripción: El usuario artista en el <i>frontend</i> puede consumir varios métodos para:</p> <ul style="list-style-type: none"> • Subir canción a un álbum. • Visualizar canciones. • Actualizar canción. • Eliminar canción. 	
<p>Observación:</p> <p>Solo pueden subir una canción por solicitud, dentro de un álbum creado por el artista, además haber enviado una solicitud y la misma ser aceptada.</p>	

TABLA XIX: Buscador de canciones.

HISTORIA DE USUARIO	
Identificador: HU010	Usuario: Artista
Nombre historia: Buscador de canciones.	
Prioridad en Negocio: Baja	Riesgo en Negocio: Baja
Iteración asignada: 4	
Responsable: Denisse Cumbal	
<p>Descripción: El usuario artista en el <i>frontend</i> tiene la posibilidad de buscar canciones.</p>	
<p>Observación:</p>	

Solo pueden subir una canción por solicitud, dentro de un álbum creado por el artista, además haber enviado una solicitud y la misma ser aceptada.

Product Backlog

La **TABLA XX** muestra el orden de prioridad de cada módulo que se ha implementado en el componente, los cuales son asignados tomando en cuenta la prioridad del usuario y su dificultad.

TABLA XX: Product Backlog.

ELABORACIÓN DEL PRODUCT BACKLOG				
ID – HU	HISTORIA DE USUARIO	ITERACIÓN	ESTADO	PRIORIDAD
HU003	Modificar perfil de usuario.	1	Finalizada	Media
HU004	Gestionar solicitudes de artistas.	2	Finalizada	Alta
HU005	Registrarse a través de un formulario.	2	Finalizada	Media
HU006	Modificar perfil de usuario y enviar solicitud.	2	Finalizada	Media
HU007	Gestionar géneros musicales.	3	Finalizada	Media
HU008	Gestionar álbumes.	3	Finalizada	Media
HU009	Gestionar canciones.	3	Finalizada	Alta
HU010	Buscador de canciones.	4	Finalizada	Baja

Sprint Backlog

La **TABLA XXI** muestra todos los *Sprint* que se han planificado en el proyecto los cuales suman un total de 5, en donde cada uno cuenta con el tiempo estimado de cada actividad para culminar con los entregables ya definidos con el usuario.

TABLA XXI: *Sprint Backlog*.

ELABORACIÓN DEL <i>SPRINT BACKLOG</i>						
ID-SB	NOMBRE	MÓDULO	ID-HU	HISTORIA DE USUARIO	TAREAS	TIEMPO ESTIMADO
SB000	Información principal del <i>Frontend</i>	Módulo – Información general	HU001	Visualizar una página informativa	<ul style="list-style-type: none"> • Diseño de prototipos. • Definir los requerimientos funcionales y no funcionales del <i>frontend</i> • Definir el menú principal. • Definir la información que se va visualizar en los diferentes módulos informativos. • Pruebas respectivas. 	25h

SB001	Inicio de sesión y recuperación de contraseña del usuario	Módulo – Inicio de Sesión rol administrador	HU002	Iniciar sesión, cerrar sesión y recuperar contraseña	<ul style="list-style-type: none"> • Diseño de prototipos. • Definir los campos obligatorios para iniciar sesión y recuperar contraseña • Diseño de interfaz para el inicio de sesión y recuperar contraseña. • Crear las funciones para el inicio de sesión y recuperar contraseña del usuario administrador en el <i>frontend</i>. • Validación de los campos en iniciar sesión y recuperar contraseña. • Consumir varios métodos para iniciar sesión como administrador. • Pruebas con el usuario administrador en el <i>frontend</i>. 	25h
		Módulo – Inicio de Sesión rol Artista		Iniciar sesión, cerrar sesión y recuperar contraseña	<ul style="list-style-type: none"> • Diseño de prototipos. • Definir los campos obligatorios para iniciar sesión y recuperar contraseña • Diseño de interfaz para el inicio de sesión y recuperar contraseña. • Crear las funciones para el inicio de sesión y recuperar contraseña del usuario administrador en el <i>frontend</i>. • Validación de los campos en iniciar sesión y recuperar contraseña. • Consumir varios métodos para iniciar sesión como artista. • Pruebas con el usuario artista en el <i>frontend</i>. 	

	Modificar - Perfil administrador	Módulo – Modificar Perfil rol administrador	HU003		<ul style="list-style-type: none"> • Diseño de prototipos. • Definir los campos que se mostraran en el módulo • Definir los campos que serán editables • Diseño de interfaz para el perfil del usuario administrador. • Consumir varios métodos para modificar el perfil • Pruebas con el usuario administrador en el <i>frontend</i>. 	
SB002	Solicitudes de artistas	Modulo – Solicitud de artistas	HU004	Gestionar solicitudes de artistas	<ul style="list-style-type: none"> • Diseño de prototipos. • Definir los campos de visualización de las solicitudes de los artistas. • Definir los campos que serán editables. • Diseño de interfaz para las solicitudes de los artistas del usuario administrador. • Consumir varios métodos para visualizar las solicitudes. • Pruebas con el usuario administrador en el <i>frontend</i>. 	
		Modulo – Solicitud de artistas aprobadas			<ul style="list-style-type: none"> • Diseño de prototipos. • Definir los campos de visualización de las solicitudes de los artistas. • Definir los campos que serán editables. • Diseño de interfaz para las solicitudes de los artistas del usuario administrador. 	

					<ul style="list-style-type: none"> • Consumir varios métodos para visualizar y aprobar las solicitudes. • Pruebas con el usuario administrador en el <i>frontend</i>. 	40h
					<ul style="list-style-type: none"> • Diseño de prototipos. • Definir los campos de visualización de las solicitudes de los artistas. • Definir los campos que serán editables. • Diseño de interfaz para las solicitudes de los artistas del usuario administrador. • Consumir varios métodos para visualizar y rechazar las solicitudes. • Pruebas con el usuario administrador en el <i>frontend</i>. 	
Registrarse	Modulo – Registrarse rol artista	HU005	Registrarse a través de un formulario	<ul style="list-style-type: none"> • Diseño de prototipos. • Definir los campos para el registro del usuario artista. • Diseño de interfaz para las solicitudes de los artistas del usuario administrador. • Consumir varios métodos para registrarse como artista. • Pruebas con el usuario artista en el <i>frontend</i>. 		

	Modificar - Perfil artista y enviar solicitud	Módulo – Modificar perfil rol artista y enviar solicitud	HU006	Modificar perfil de usuario.	<ul style="list-style-type: none"> • Diseño de prototipos. • Definir los campos que se mostraran en el módulo. • Definir los campos que serán editables. • Diseño de interfaz para el perfil del usuario artista y de enviar solicitud • Consumir varios métodos para editar el perfil artista y enviar solicitud. • Pruebas con el usuario artista en el <i>frontend</i>. 	
SB003	Géneros musicales	Módulo – Crear Géneros musicales	HU007	Gestionar géneros musicales	<ul style="list-style-type: none"> • Diseño de prototipos. • Definir los campos para crear los géneros musicales • Diseño de interfaz para crear géneros musicales. • Consumir varios métodos para crear géneros musicales. • Pruebas con el usuario artista en el <i>frontend</i> 	65h
		Módulo – Visualizar Géneros musicales			<ul style="list-style-type: none"> • Diseño de prototipos. • Definir los campos para visualizar los géneros musicales • Diseño de interfaz para visualizar géneros musicales. • Consumir varios métodos para visualizar géneros musicales. • Pruebas con el usuario artista en el <i>frontend</i> 	
		Módulo – Eliminar géneros musicales			<ul style="list-style-type: none"> • Diseño de prototipos. • Definir los campos que serán eliminados de los géneros musicales • Diseño de interfaz para editar géneros musicales. • Consumir varios métodos para eliminar géneros musicales. 	

					<ul style="list-style-type: none"> • Pruebas con el usuario artista en el <i>frontend</i> 	
Álbumes	Módulo – Crear álbumes	HU008	Gestionar álbumes.	<ul style="list-style-type: none"> • Diseño de prototipos. • Definir los campos para crear los álbumes. • Diseño de interfaz para crear los álbumes. • Consumir varios métodos para crear álbumes musicales. • Pruebas con el usuario artista en el <i>frontend</i> 		
	Módulo – Visualizar álbumes			<ul style="list-style-type: none"> • Definir los campos para visualizar los álbumes. • Diseño de interfaz para visualizar álbumes. • Consumir varios métodos para visualizar álbumes musicales. • Pruebas con el usuario artista en el <i>frontend</i> 		
	Módulo – Eliminar álbumes			<ul style="list-style-type: none"> • Diseño de prototipos. • Definir los campos para eliminar los álbumes. • Diseño de interfaz para eliminar álbumes. • Consumir varios métodos para eliminar álbumes musicales. • Pruebas con el usuario artista en el <i>frontend</i> 		
	Módulo – Subir canción			<ul style="list-style-type: none"> • Diseño de prototipos. • Definir los campos para subir las canciones. • Diseño de interfaz para subir las canciones. 		

	Canciones		HU009	Gestionar canciones.	<ul style="list-style-type: none"> • Consumir varios métodos para subir canciones • Pruebas con el usuario artista en el <i>frontend</i> 	
		Módulo – visualizar canción			<ul style="list-style-type: none"> • Diseño de prototipos. • Definir los campos para visualizar las canciones. • Diseño de interfaz para visualizar las canciones. • Consumir varios métodos para visualizar canciones • Pruebas con el usuario artista en el <i>frontend</i>. 	
		Módulo – editar y eliminar canción			<ul style="list-style-type: none"> • Diseño de prototipos. • Definir los campos para editar los álbumes. • Diseño de interfaz para editar y eliminar álbumes. • Consumir varios métodos para editar y eliminar canciones • Pruebas con el usuario artista en el <i>frontend</i> 	
SB004	Buscador	Módulo – Buscador de canciones	HU009	Buscador de canciones	<ul style="list-style-type: none"> • Diseño de prototipos. • Diseño de interfaz para editar y eliminar canciones. • Pruebas con el usuario artista en el <i>frontend</i> 	5h
SB005	Pruebas y	Pruebas en el <i>frontend</i>	<ul style="list-style-type: none"> • Pruebas en el <i>frontend</i>. • Pruebas de carga. 			

	despliegue del <i>frontend</i>		<ul style="list-style-type: none"> • Pruebas de compatibilidad. • Pruebas de aceptación. 	40h
		Publicación del <i>frontend</i>	<ul style="list-style-type: none"> • Despliegue del <i>frontend</i> en <i>Netlify</i> 	
Documentación			<ul style="list-style-type: none"> • Trabajo de Integración Curricular. • Anexos. 	40h
TOTAL				240

Diseño de Interfaces

A continuación, se visualiza los diferentes *mockups* de cada uno de los módulos en el componente del *frontend*, donde se representa cada uno de los diseños para cada interfaz. Los cuales van desde la **Fig.26** a la **Fig. 42**.

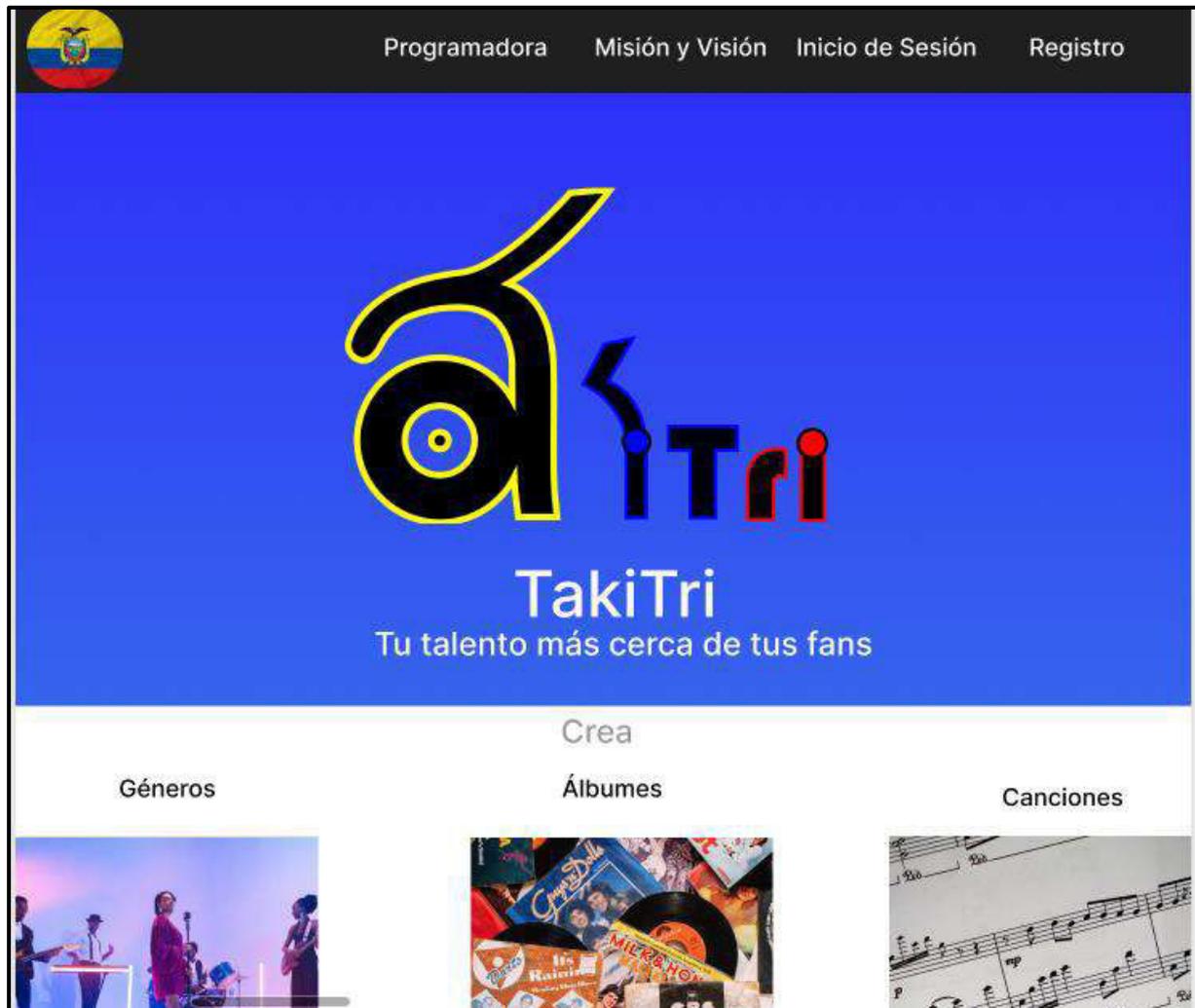


Fig.26: Visualizar página Informativa.

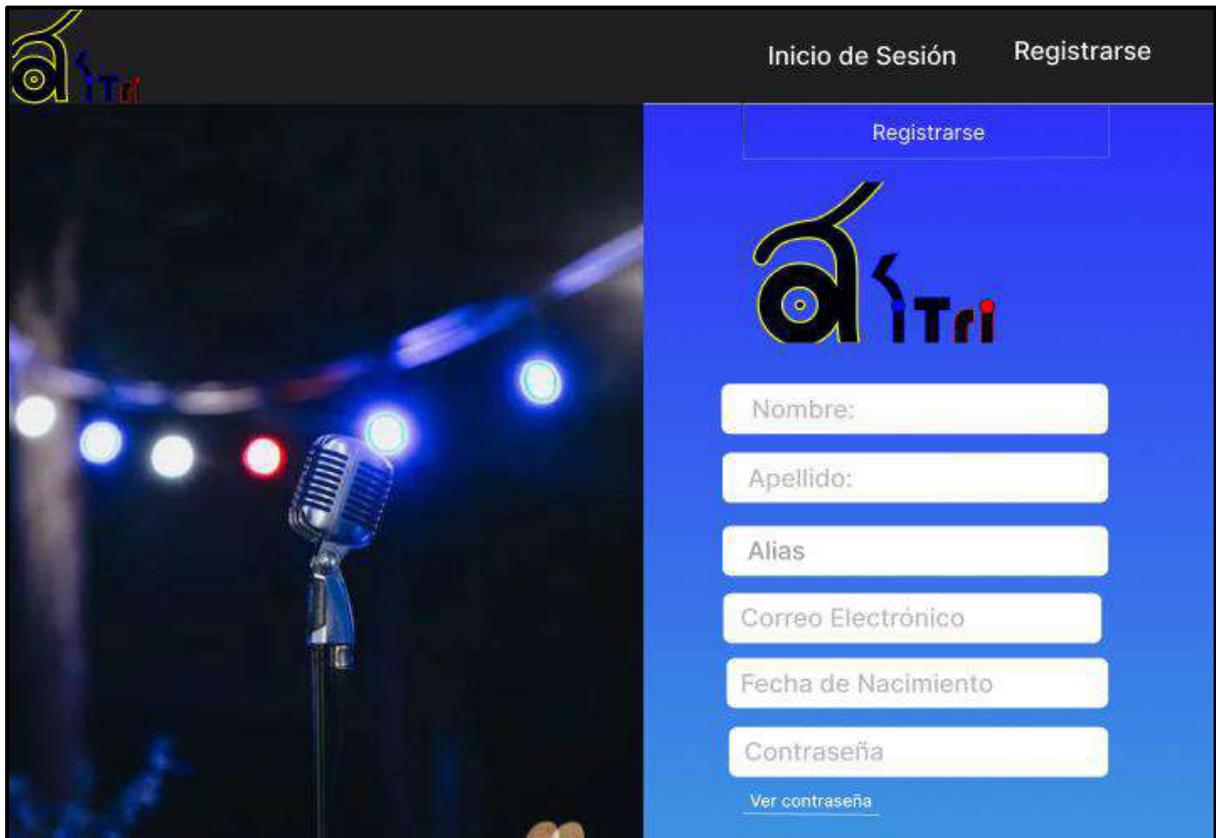


Fig. 27: Registrarse.

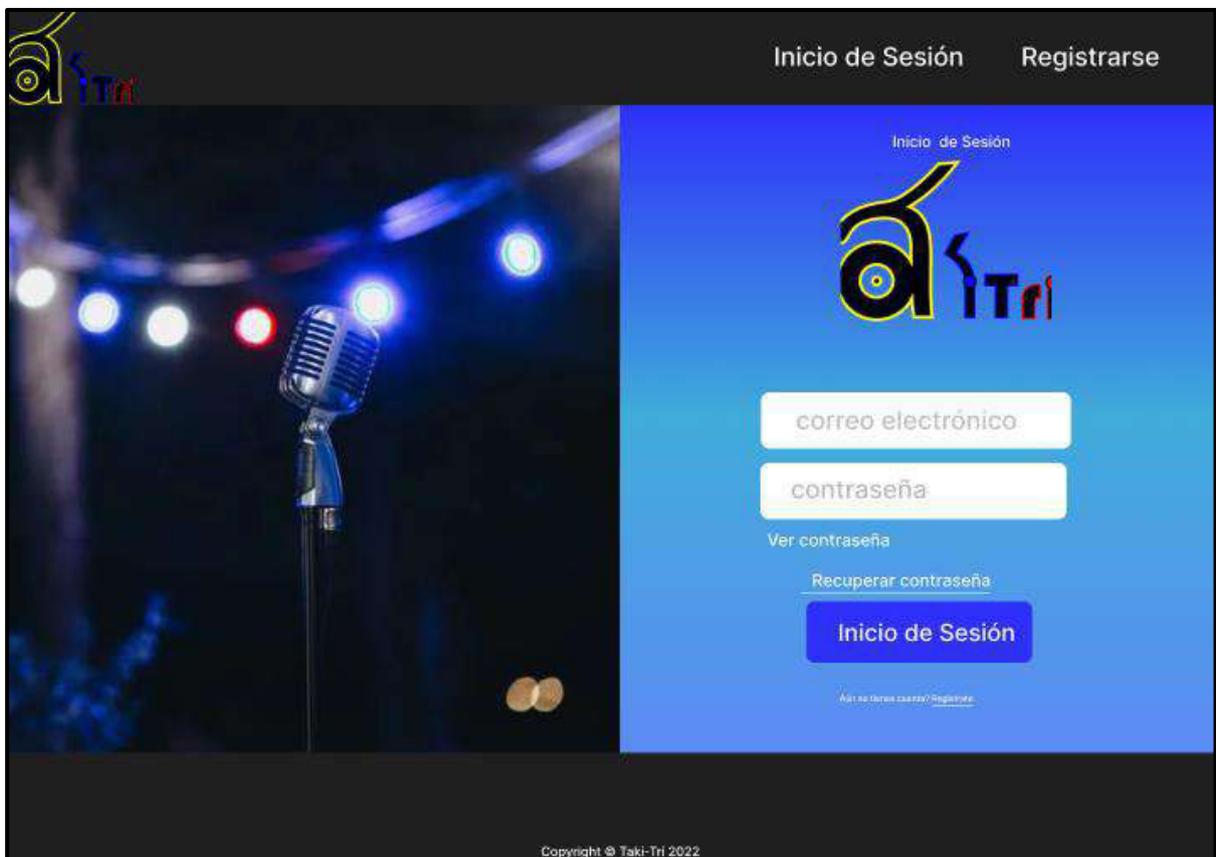


Fig. 28: Inicio de Sesión.

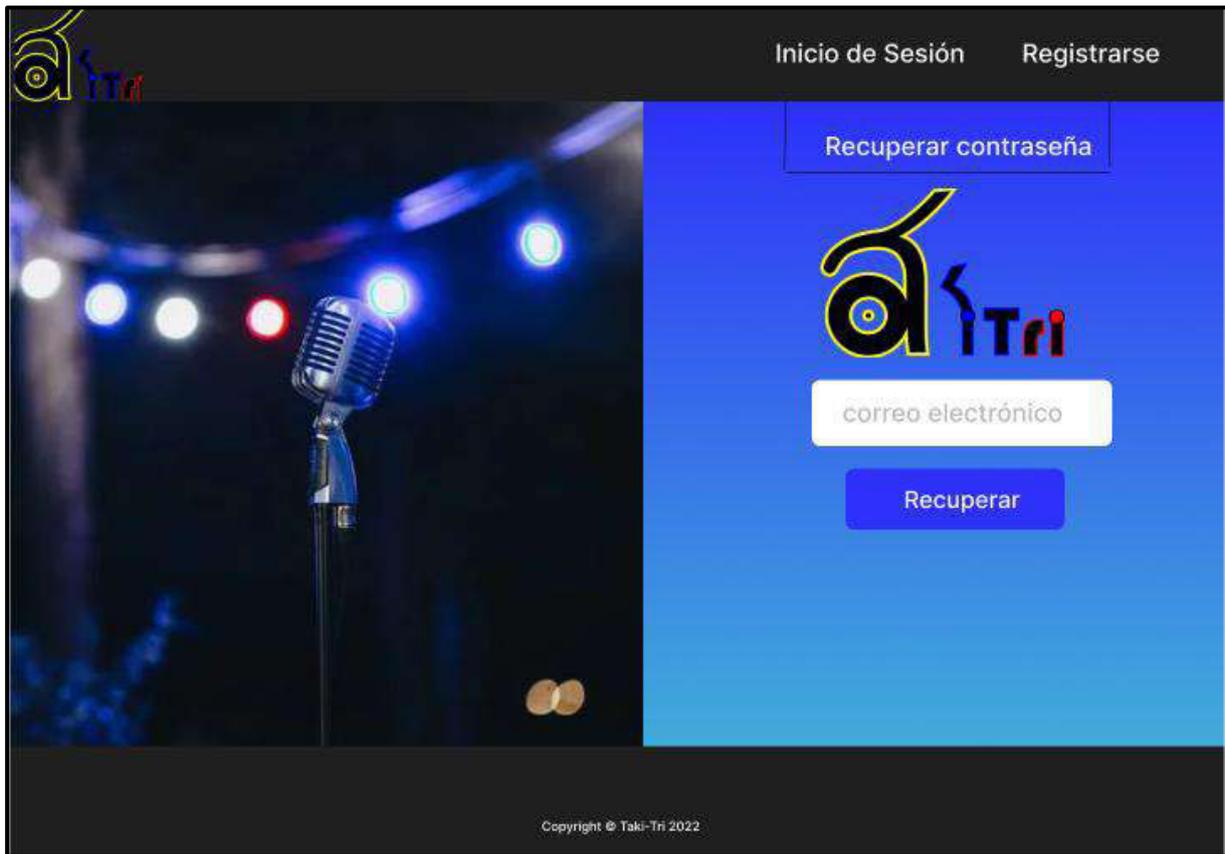


Fig. 29:Recuperar contraseña.

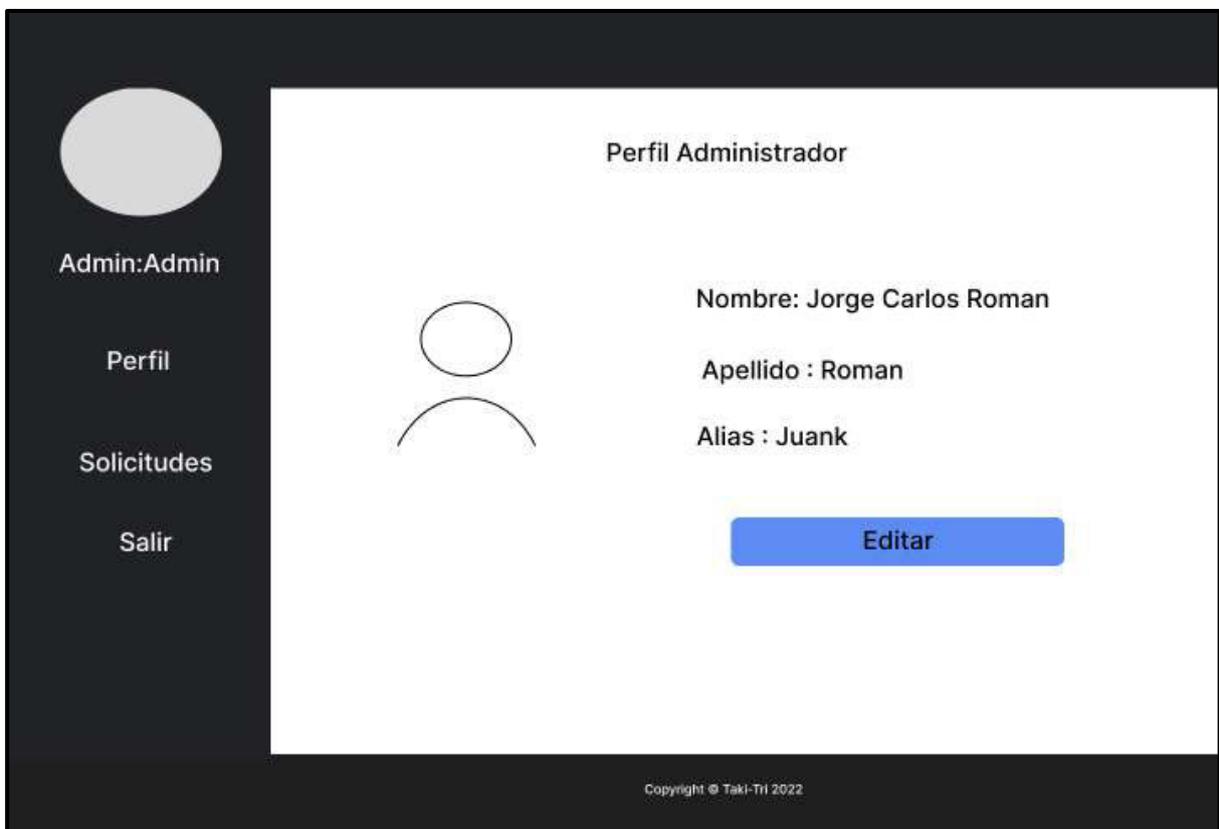


Fig. 30: Módulo Perfil administrador /artista.

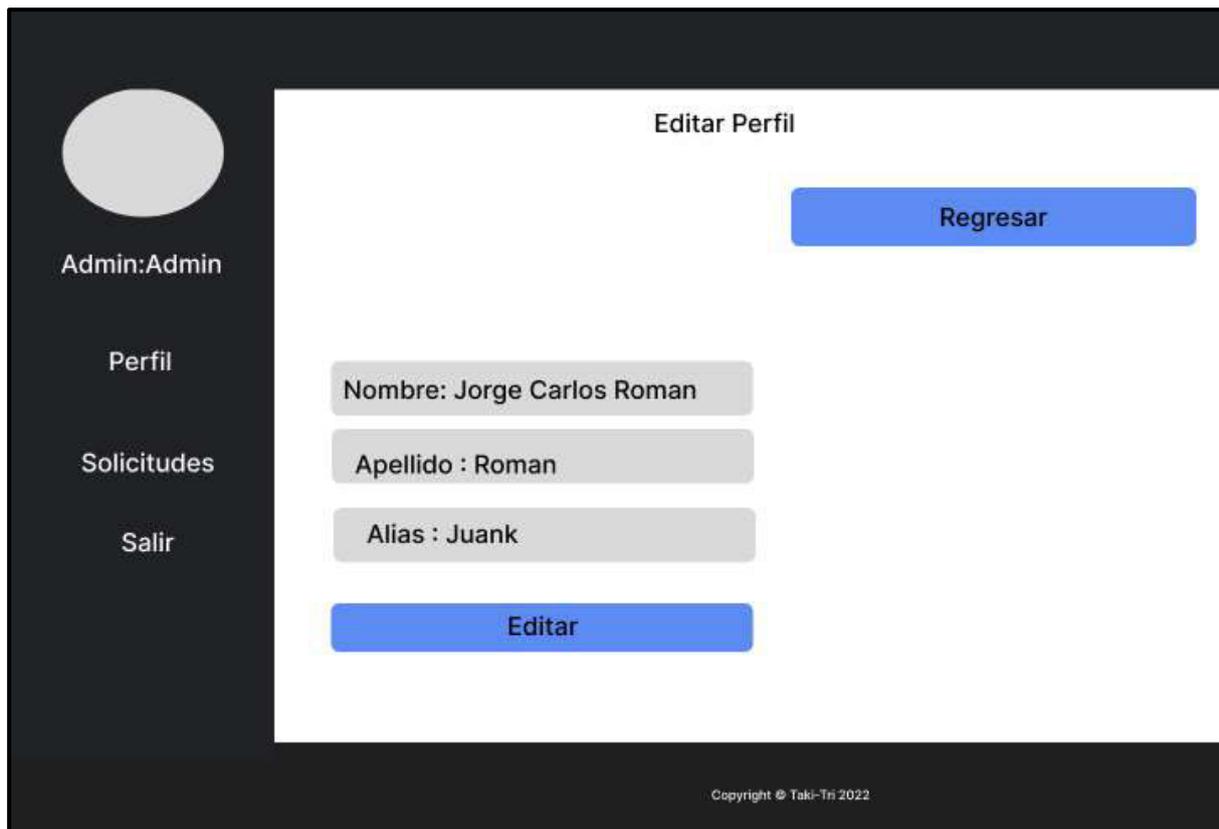


Fig. 31: Módulo editar perfil administrador / artista.



Fig. 32: Módulo de Gestionamiento solicitudes usuario Administrador.



Fig. 33: Módulo de solicitudes aceptadas.



Fig. 34: Módulo de solicitudes rechazadas.



Fig. 35: Módulo Enviar Solicitud usuario artista.

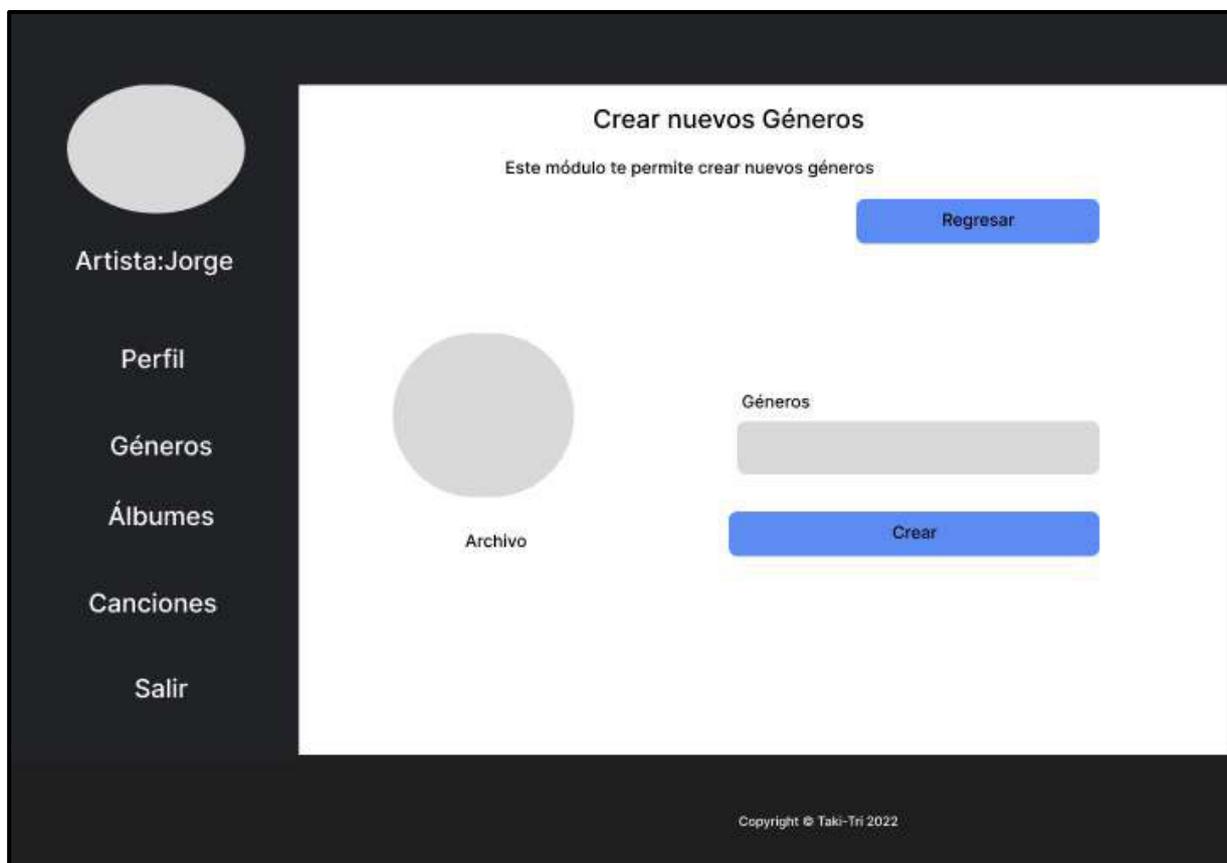


Fig. 36: Módulo Crear géneros.

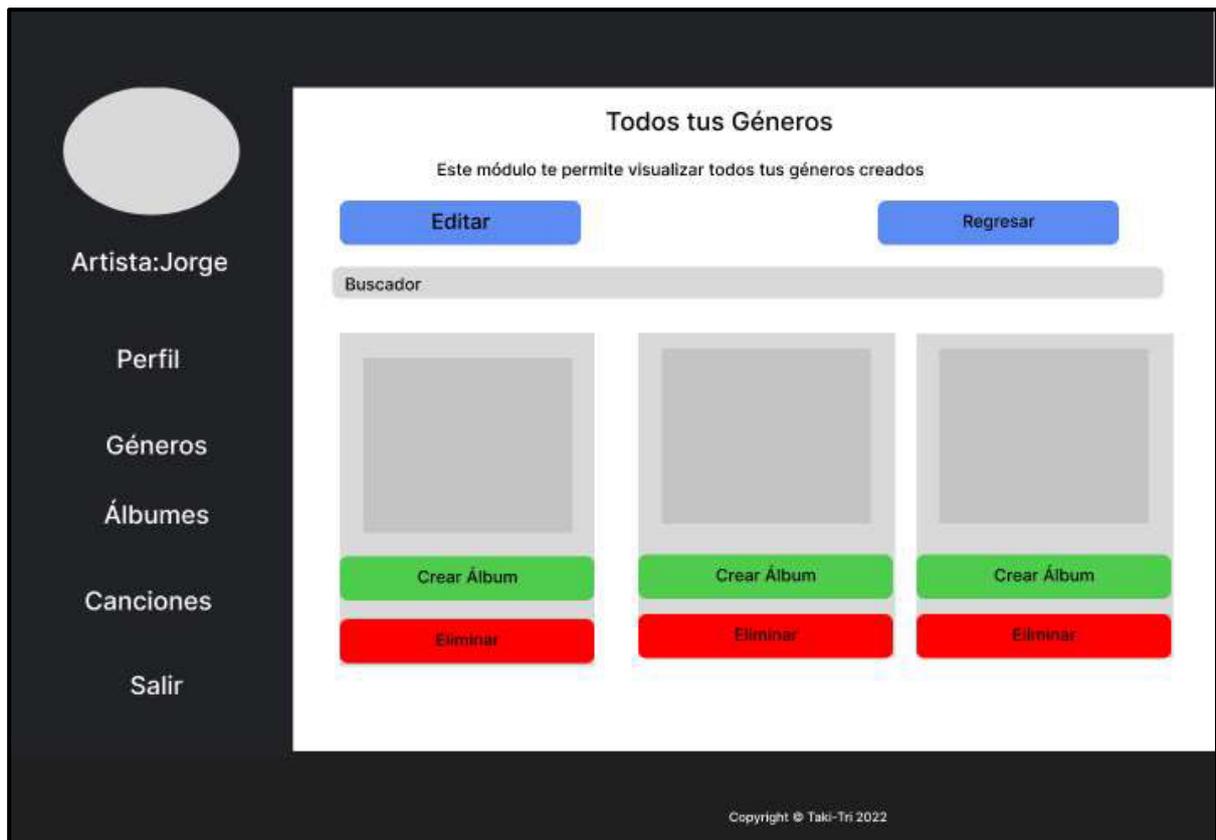


Fig. 37: Módulo Visualizar géneros creados.

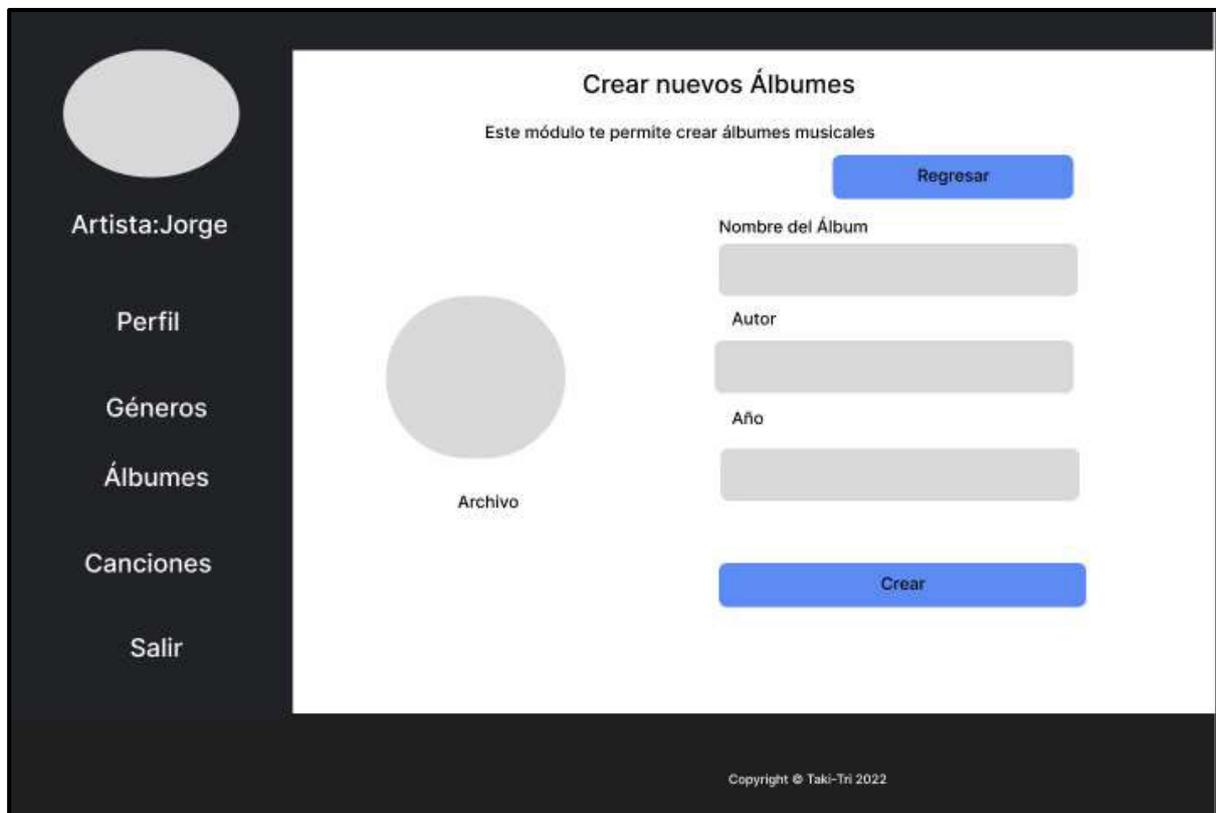


Fig. 38: Módulo Crear álbumes.

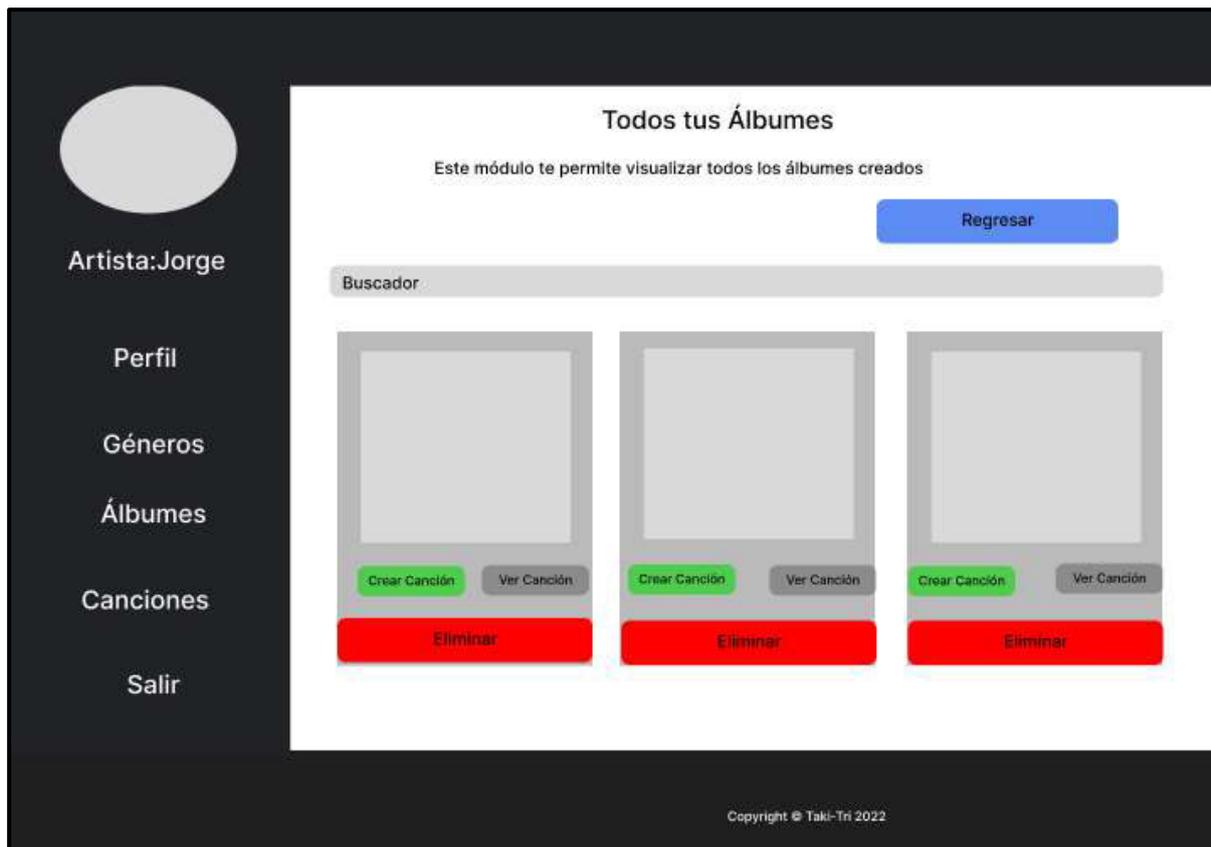


Fig. 39: Módulo visualizar los álbumes.

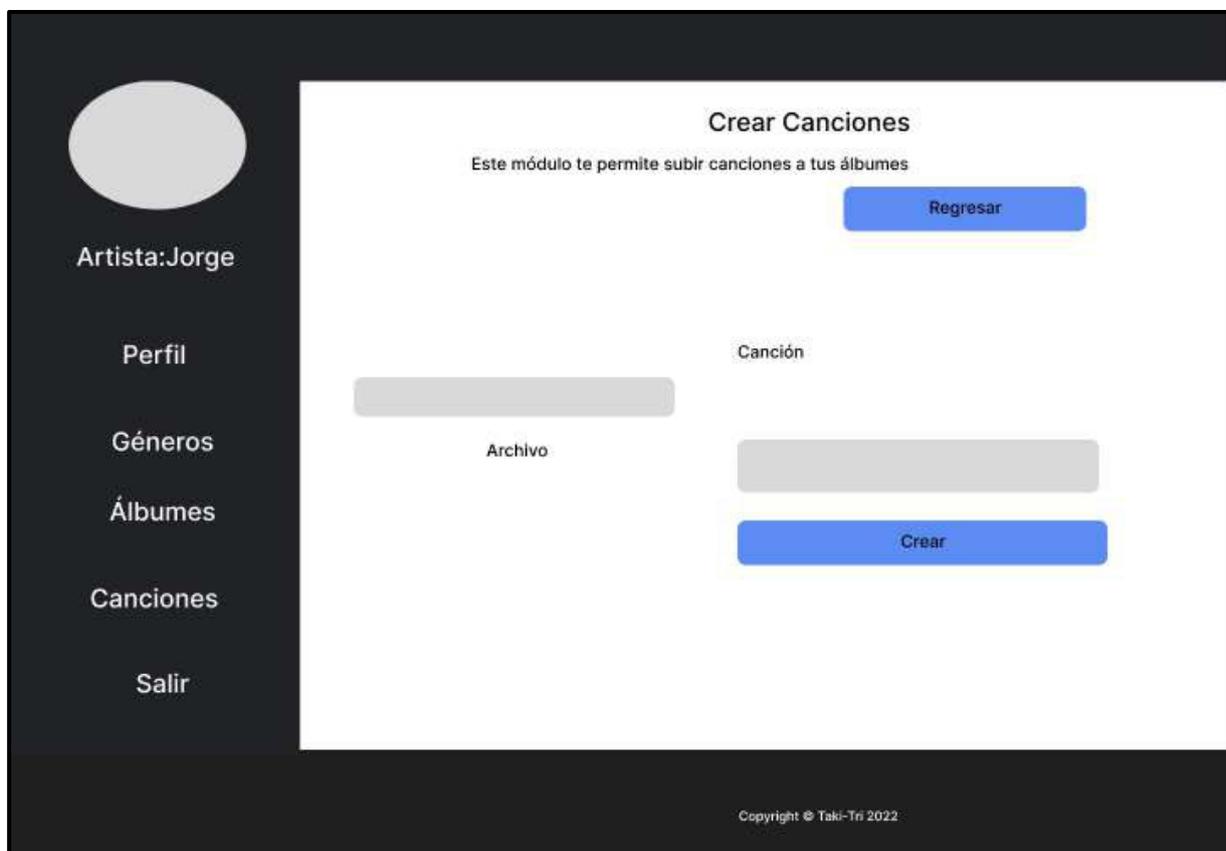


Fig. 40: Módulo crear canciones.

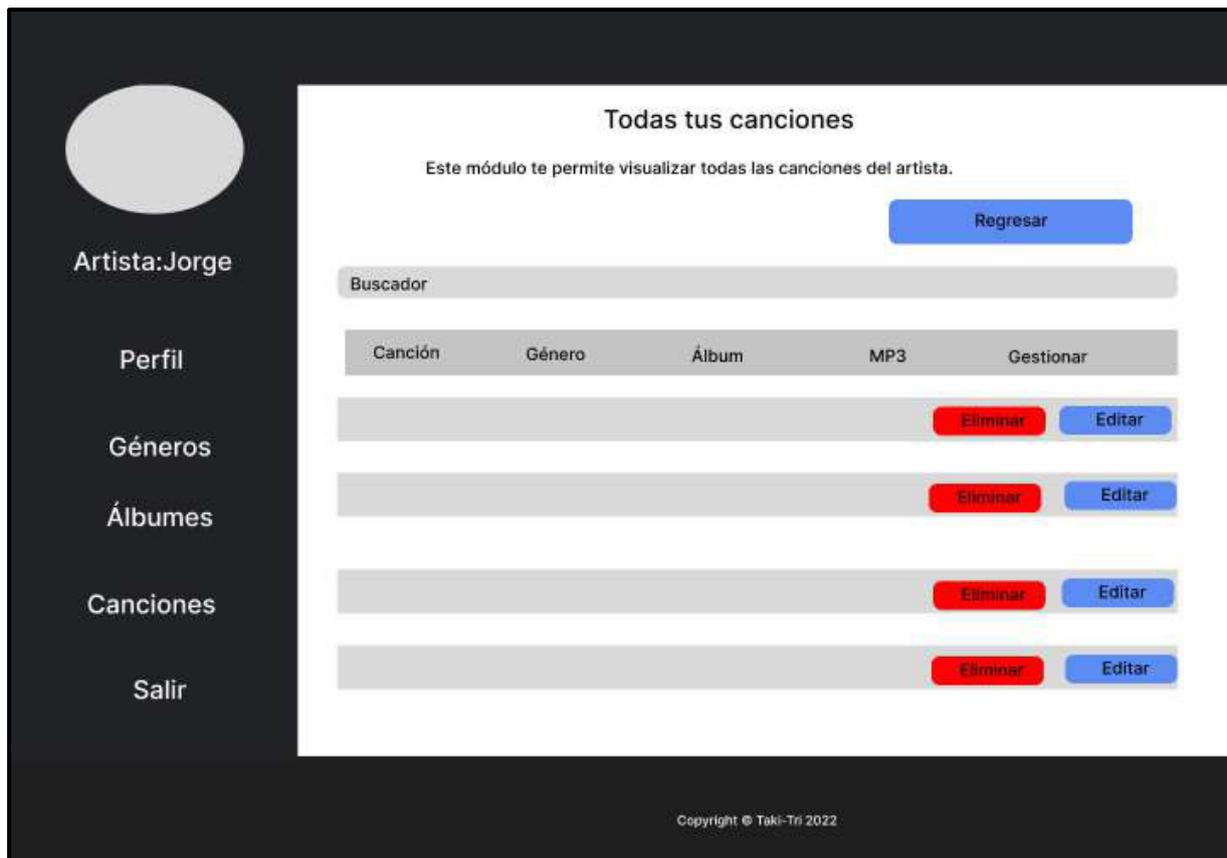


Fig. 41: Módulo visualizar canciones.

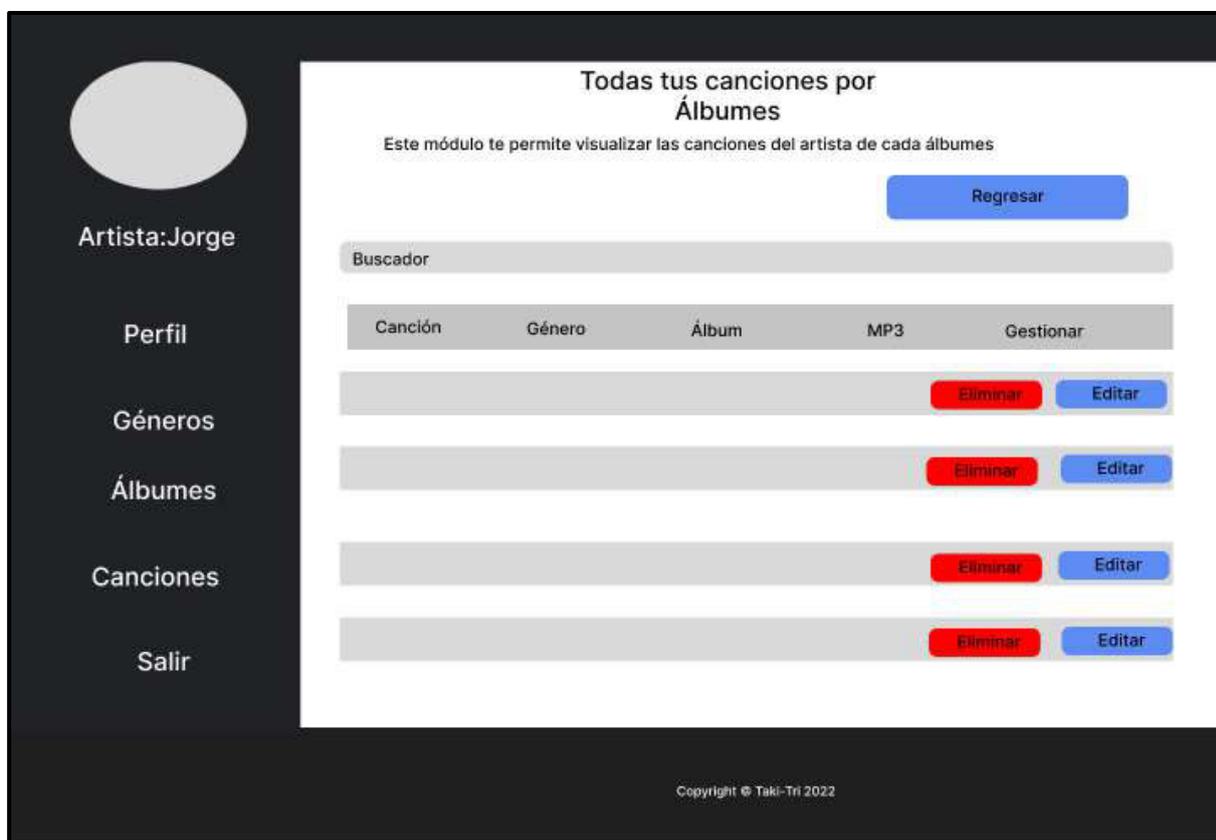


Fig. 42: Módulo buscador de canciones.

Pruebas

Culminada la etapa de codificación se ha desarrollado la implementación de diversas pruebas como: unitarias, compatibilidad y de aceptación para verificar la eficacia del código del componente del *frontend* de cada módulo.

Pruebas unitarias

En la **Fig. 43** se visualiza el comando en karma para la realización de las pruebas unitarias de las principales funcionalidades del *frontend*, donde se muestra si el módulo al ser sometido a su respectiva ejecución pasa el *testing*, como se visualiza desde la **Fig. 43** a la **Fig. 53**.

```
PS C:\Users\CUMBAL\Desktop\tesis2\noga_5_semestre> ng test
Your global Angular CLI version (14.0.0) is greater than your local version (13.2.6). The local Angular CLI version is used.

To disable this warning use "ng config -g cli.warnings.versionMismatch false".
✓ Browser application bundle generation complete.
28 07 2022 00:52:57.014:WARN [karma]: No captured browser, open http://localhost:9876/
28 07 2022 00:52:57.112:INFO [karma-server]: Karma v6.3.17 server started at http://localhost:9876/
```

Fig. 43: Prueba unitaria Comando para su ejecución.

```
import { TestBed } from '@angular/core/testing';
import { RouterTestingModule } from '@angular/router/testing';
import { AppComponent } from './app.component';

describe('AppComponent', () => {
  beforeEach(async () => {
    await TestBed.configureTestingModule({
      imports: [
        RouterTestingModule
      ],
      declarations: [
        AppComponent
      ],
    }).compileComponents();
  });

  it('Funcionalidad de Componentes', () => {
    const fixture = TestBed.createComponent(AppComponent);
    const app = fixture.componentInstance;
    expect(app).toBeTruthy();
  });
});
```

Fig. 44: Prueba unitaria *funcionalidad entre Componente#2*.

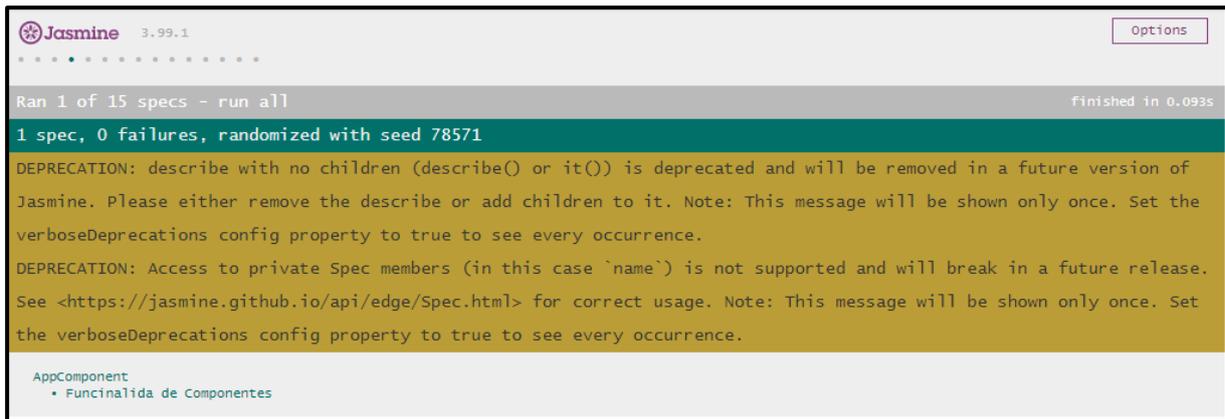


Fig. 45: Resultado de prueba funcionalidad entre Componente #2.

```
import { FiltroSolicitudRechazadasPipe } from './filtro-solicitud-rechazadas.pipe';

describe('FiltroSolicitudRechazadasPipe', () => {
  it('create an instance', () => {
    const pipe = new FiltroSolicitudRechazadasPipe();
    expect(pipe).toBeTruthy();
  });
});
```

Fig. 46: Prueba unitaria Filtro de Solicitudes Rechazadas#3.

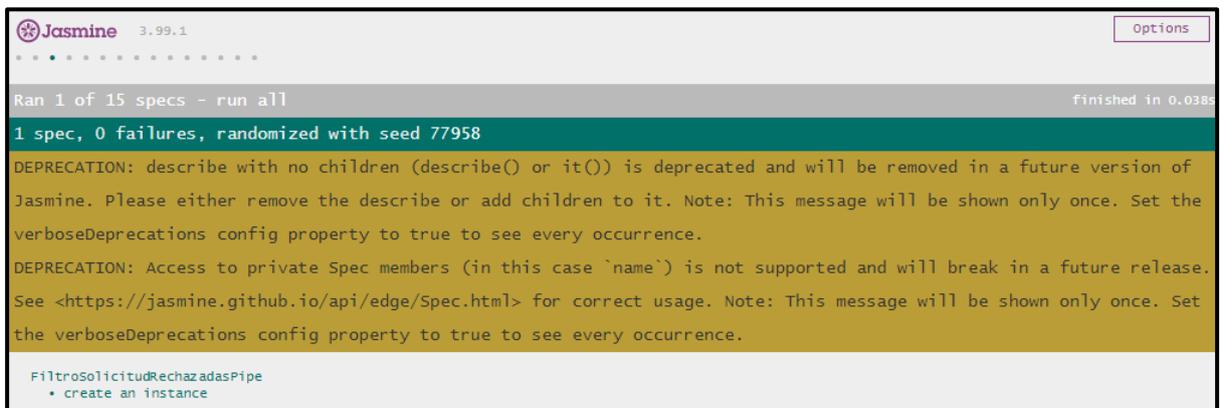


Fig. 47: Prueba unitaria Filtro de Solicitudes Rechazadas#3.

```

import { ComponentFixture, TestBed } from "@angular/core/testing"

import { GenerosService } from '../generos.service';

describe('Pruebas Genero mio', () =>{
  let fixture: ComponentFixture<GenerosService>;
  let component: GenerosService;

  it('Prueba de Genero', () =>{

    fixture = TestBed.createComponent(GenerosService);
    component = fixture.componentInstance;
    const datos_generos = {name: "Pasillo",
imageURL:"https://firebasestorage.googleapis.com/v0/b/takitesis.appspot.com/o/imagenGeneros%2F8d0fzRCcaJeCBk0SjI6mR
const verification = component.crearGenero(datos_generos);
expect(verification).toBeTruthy();

  })

})

```

Fig. 48: Pruebas unitarias Géneros#4.

```

Ran 1 of 15 specs - run all finished in 0.033s
1 spec, 0 failures, randomized with seed 86229
DEPRECATION: describe with no children (describe() or it()) is deprecated and will be removed in a future version of
Jasmine. Please either remove the describe or add children to it. Note: This message will be shown only once. Set the
verboseDeprecations config property to true to see every occurrence.
DEPRECATION: Access to private Spec members (in this case `name`) is not supported and will break in a future release.
See <https://jasmine.github.io/api/edge/Spec.html> for correct usage. Note: This message will be shown only once. Set
the verboseDeprecations config property to true to see every occurrence.

Pruebas Genero mio
  • Prueba de Genero

```

Fig. 49: Resultado de pruebas Géneros#4.

```

import { ComponentFixture, TestBed } from '@angular/core/testing';

import { AlbumesService } from './albumes.service';

describe('Pruebas creacion album', () =>{
  let fixture: ComponentFixture<AlbumesService>;
  let component: AlbumesService;

  it('Prueba de Album', () =>{

    fixture = TestBed.createComponent(AlbumesService);
    component = fixture.componentInstance;
    const datos_albumes = {name: "Muñequita",author:"Pamela Paez",year:"2015",
imageURL:"https://firebasestorage.googleapis.com/v0/b/takitesis.appspot.com/o/imagenAlbumes
const verification = component.crearAlbumes(datos_albumes);
expect(verification).toBeTruthy();

  })

})

```

Fig. 50: Pruebas unitarias Álbumes#5.

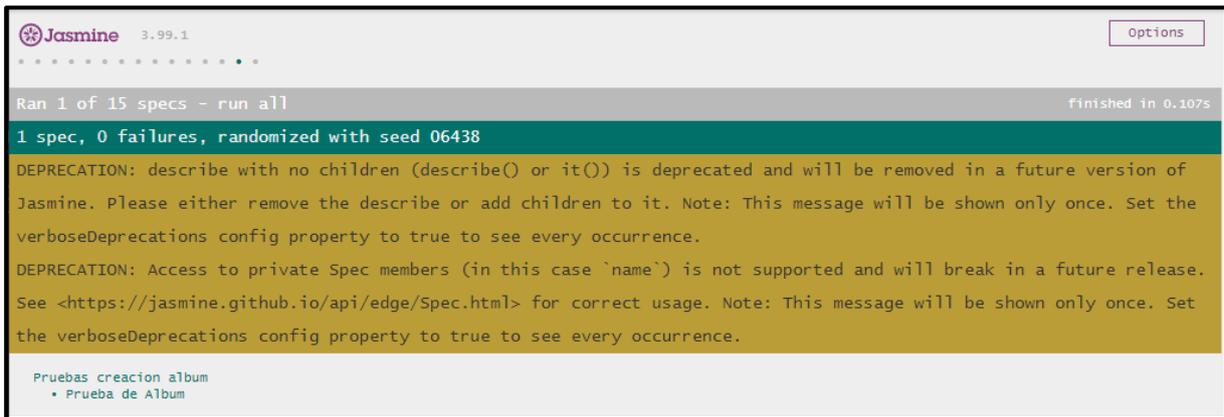


Fig. 51: Resultados de pruebas Álbumes#5.

```
describe('CancionService', () => {
  let service: CancionService;
  describe('Pruebas creacion cancion', () =>{
    let fixture: ComponentFixture<CancionService>;
    let component: CancionService;

    it('Prueba de Cancion', () =>{

      fixture = TestBed.createComponent(CancionService);
      component = fixture.componentInstance;
      const datos_cancion = { songURL: "https://firebasestorage.googleapis.com/v0/b/takitesis..",
        song_name:"Dicen",
      };
      const verification = component.creaCancion(datos_cancion);
      expect(verification).toBeTruthy();

    })
  })
});
```

Fig. 52: Prueba unitarias Canciones#6.

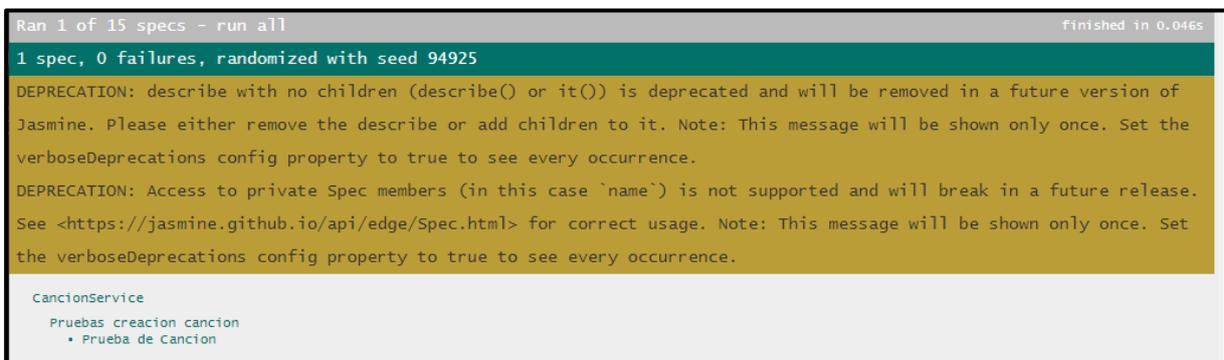


Fig. 53: Resultados de pruebas Canciones#6.

Pruebas de compatibilidad

Esta prueba, en el *frontend* se centran en comprobar las diversas funcionalidades en los principales navegadores con el objetivo de visualizar los posibles problemas al momento de presentar el contenido al usuario.

De este modo, desde la **Fig.54** hasta la **Fig.71** se visualizan los resultados que se han logrado obtener en los diferentes navegadores.

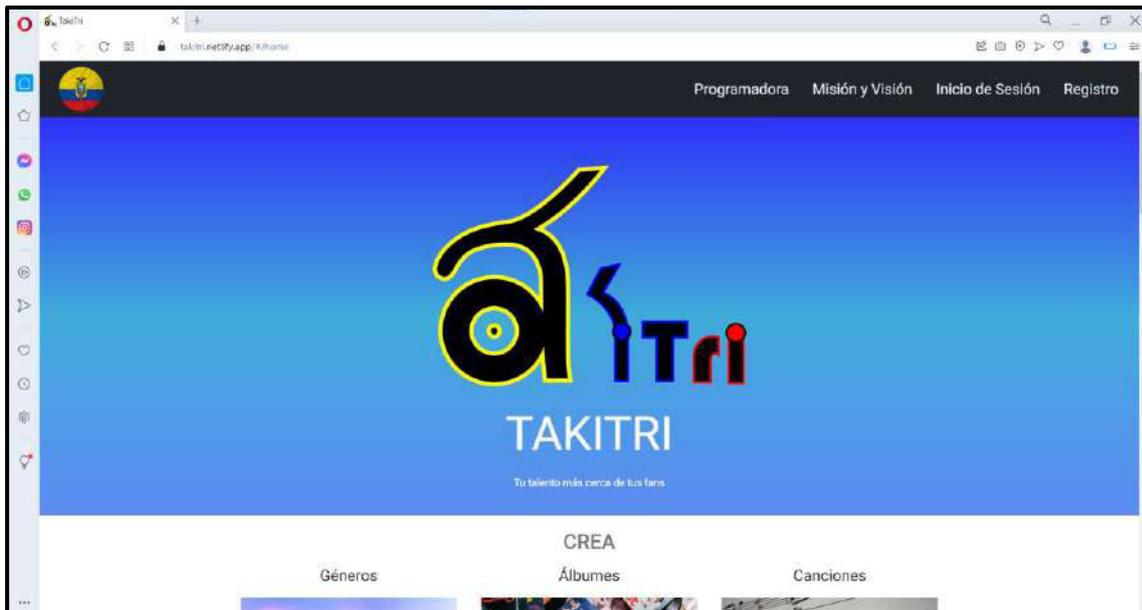


Fig.54: Página informativa vista en el navegador Opera.

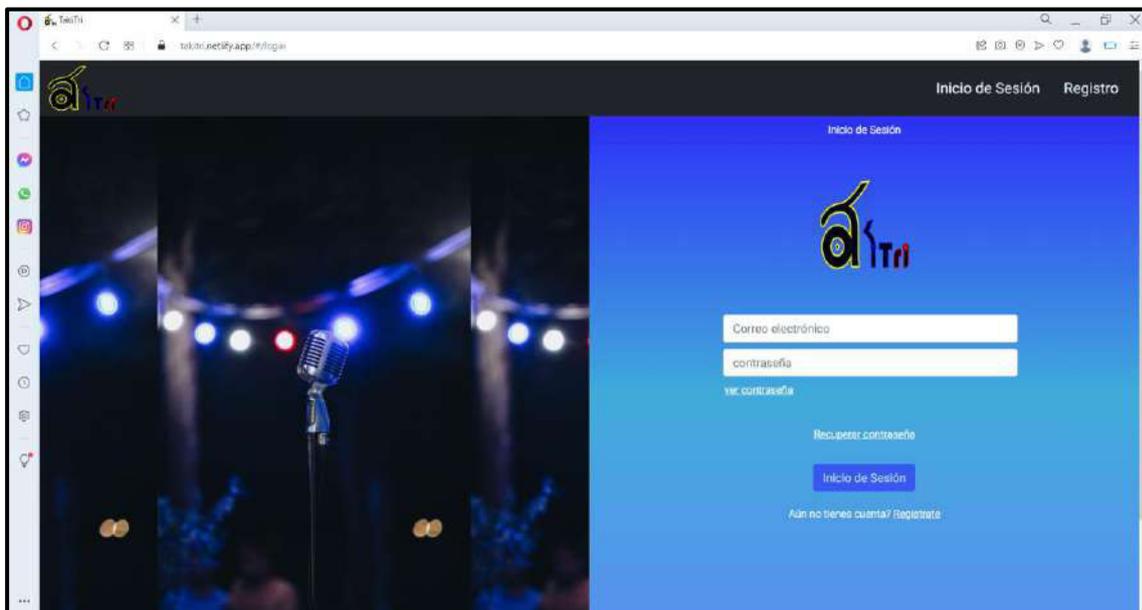


Fig. 55: Página de Inicio de Sesión en el navegador Opera.

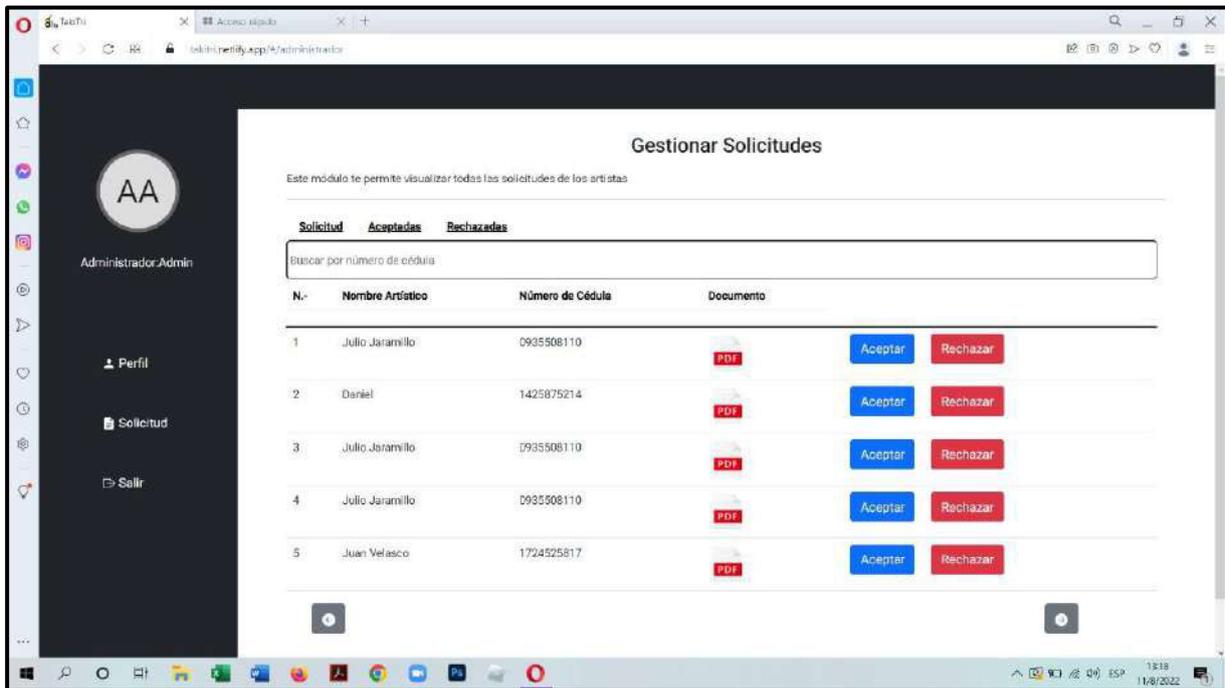


Fig. 56: Página de Solicitudes en el navegador Opera.

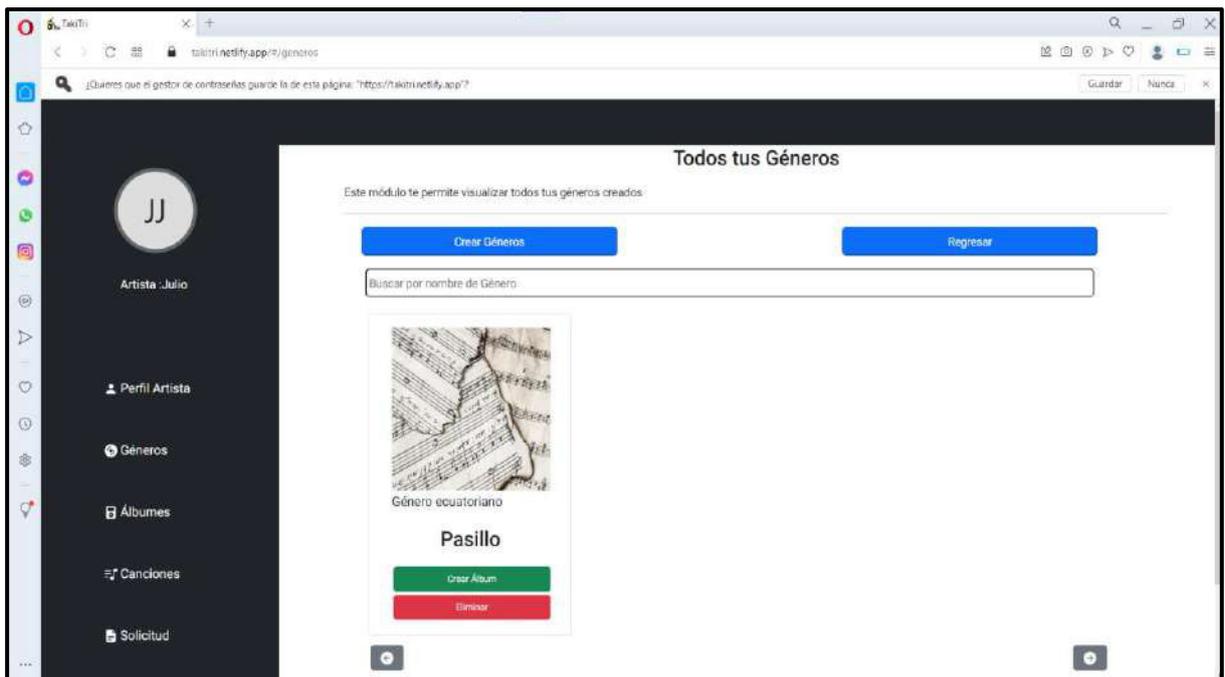


Fig. 57: Página de Géneros en el navegador Opera.

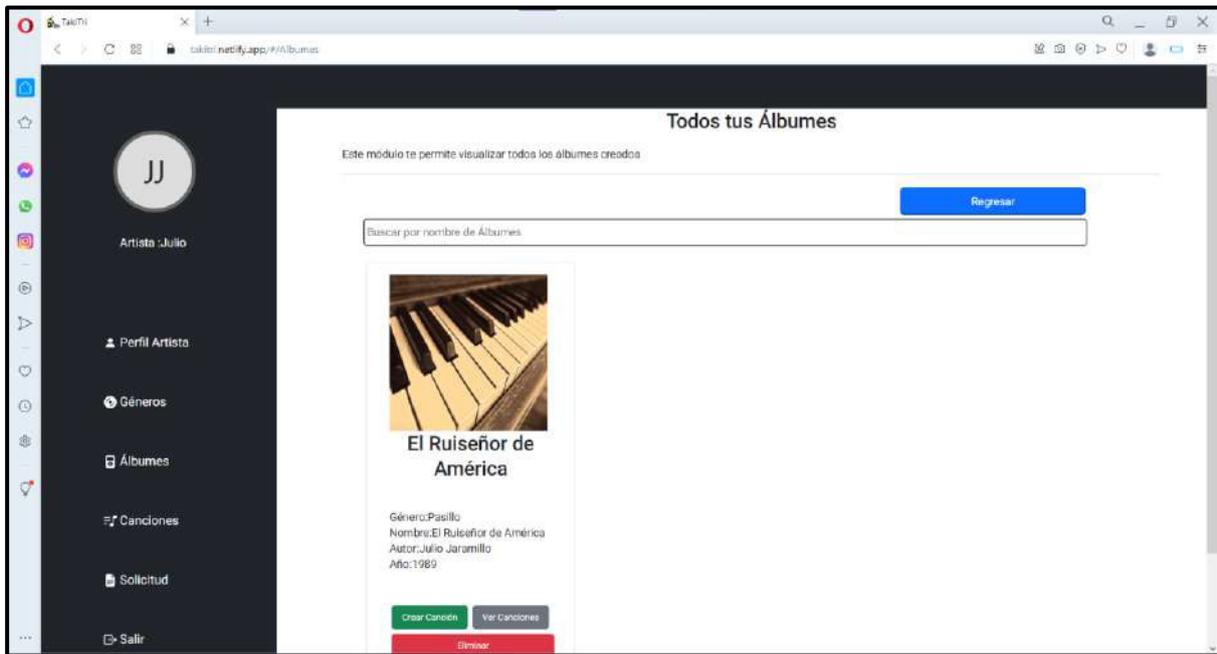


Fig. 58: Página de Álbumes en el navegador Opera.

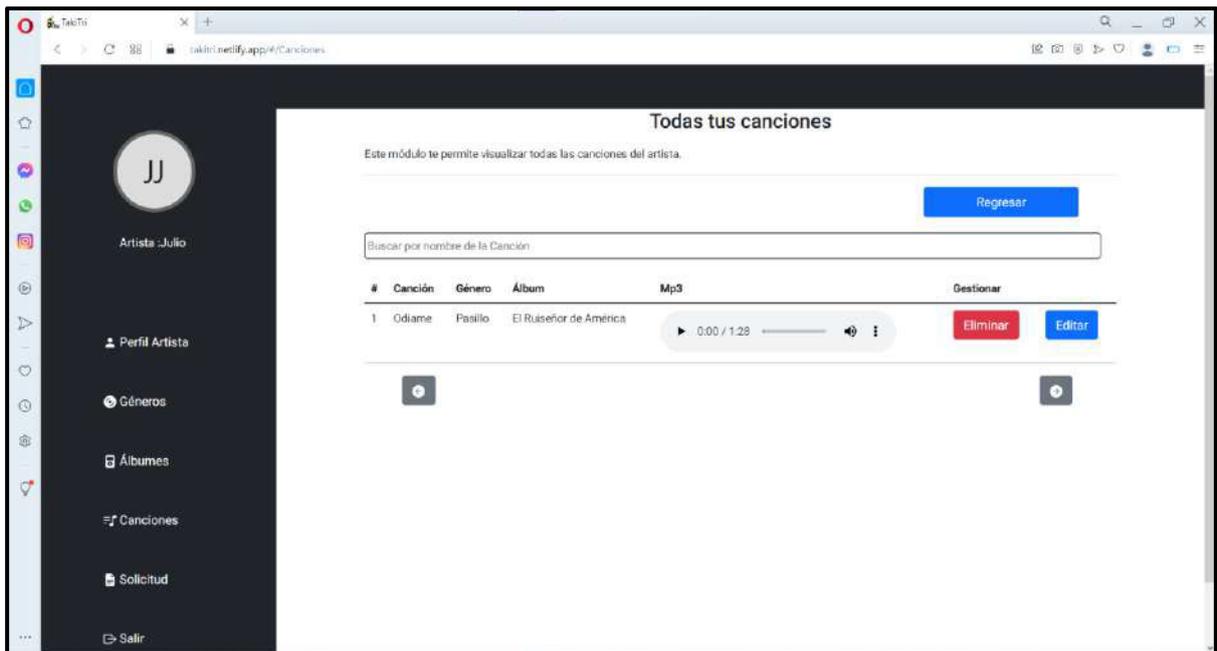


Fig. 59: Página de Canciones en el navegador Opera.

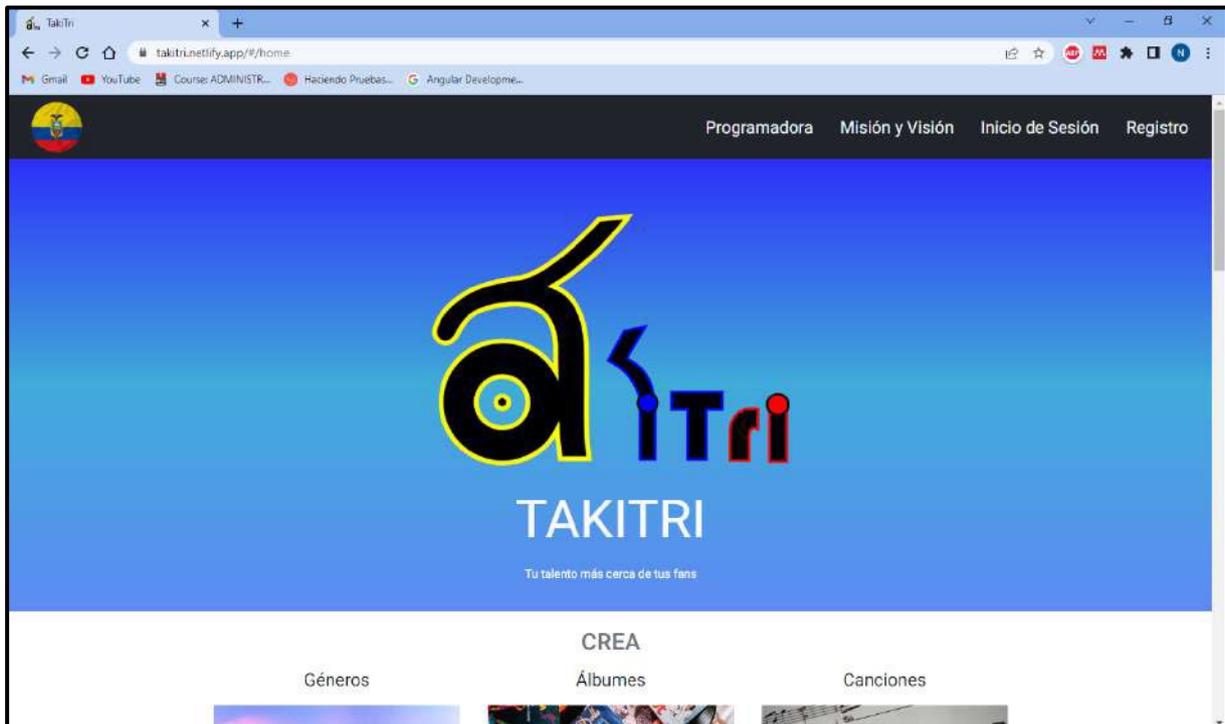


Fig. 60: Página informativa vista en el navegador *Google Chrome*.

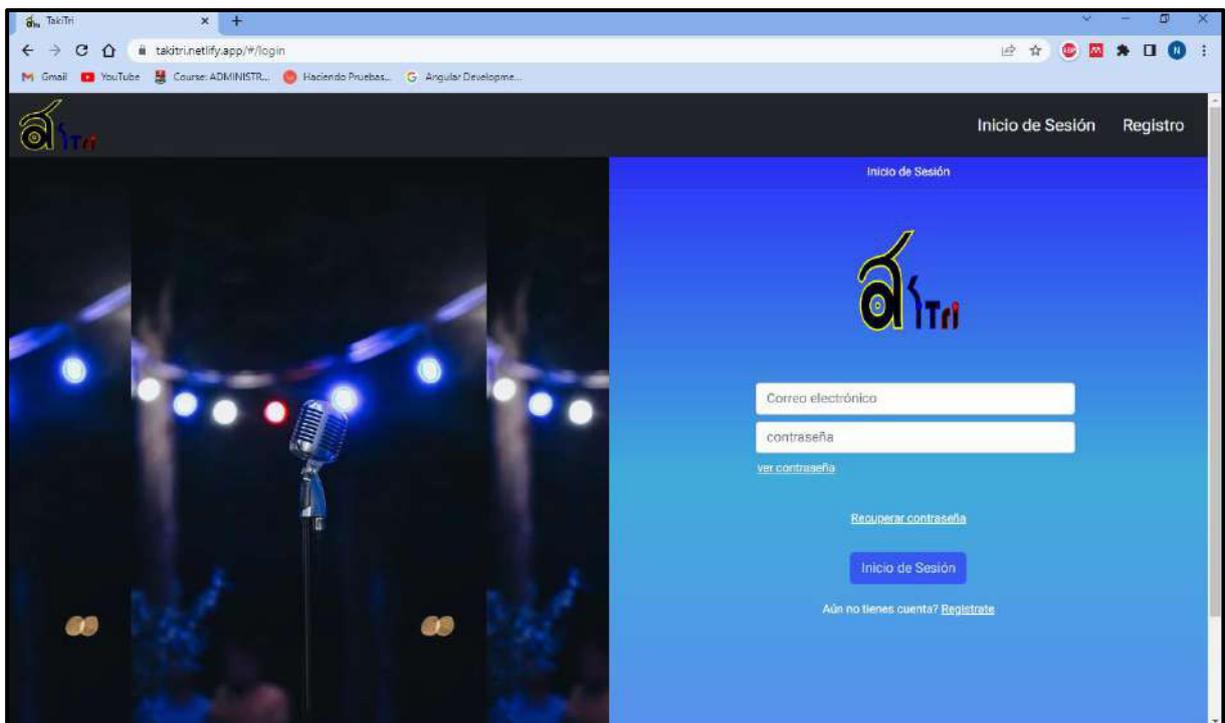


Fig. 61: Página de Inicio de Sesión en el navegador *Google Chrome*.

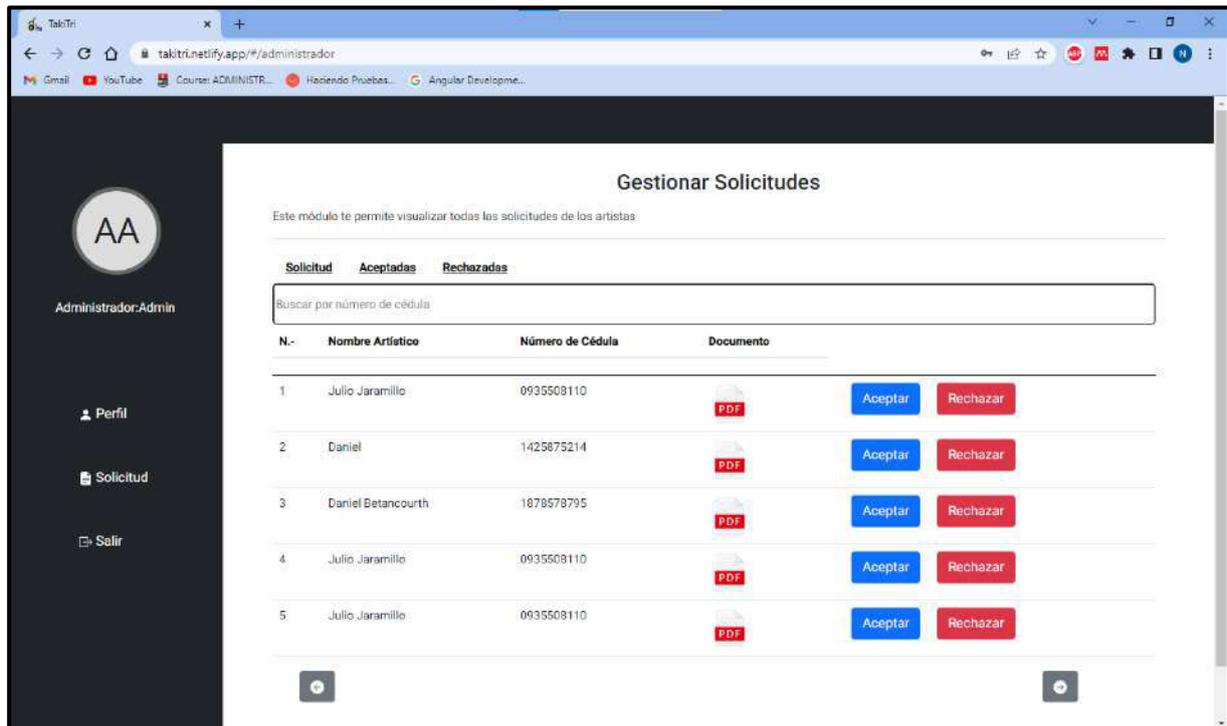


Fig. 62: Página de Solicitudes en el navegador *Google Chrome*.

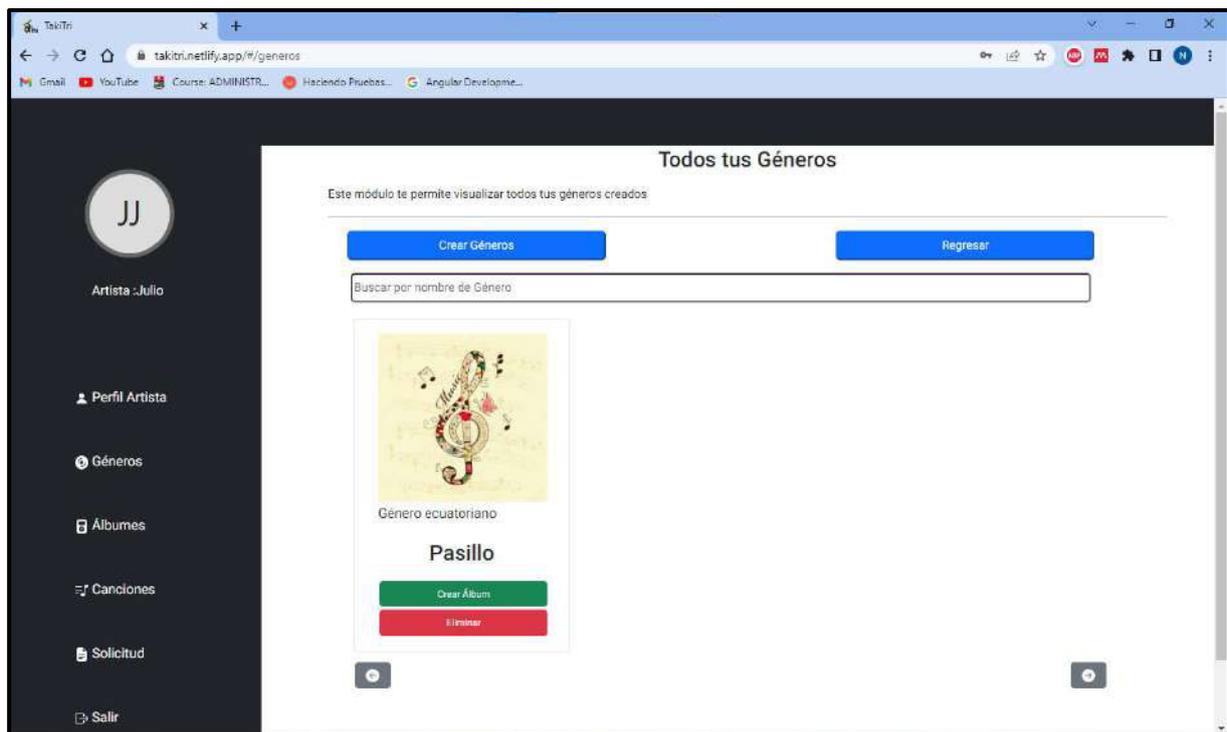


Fig.63: Página de Géneros en el navegador *Google Chrome*.

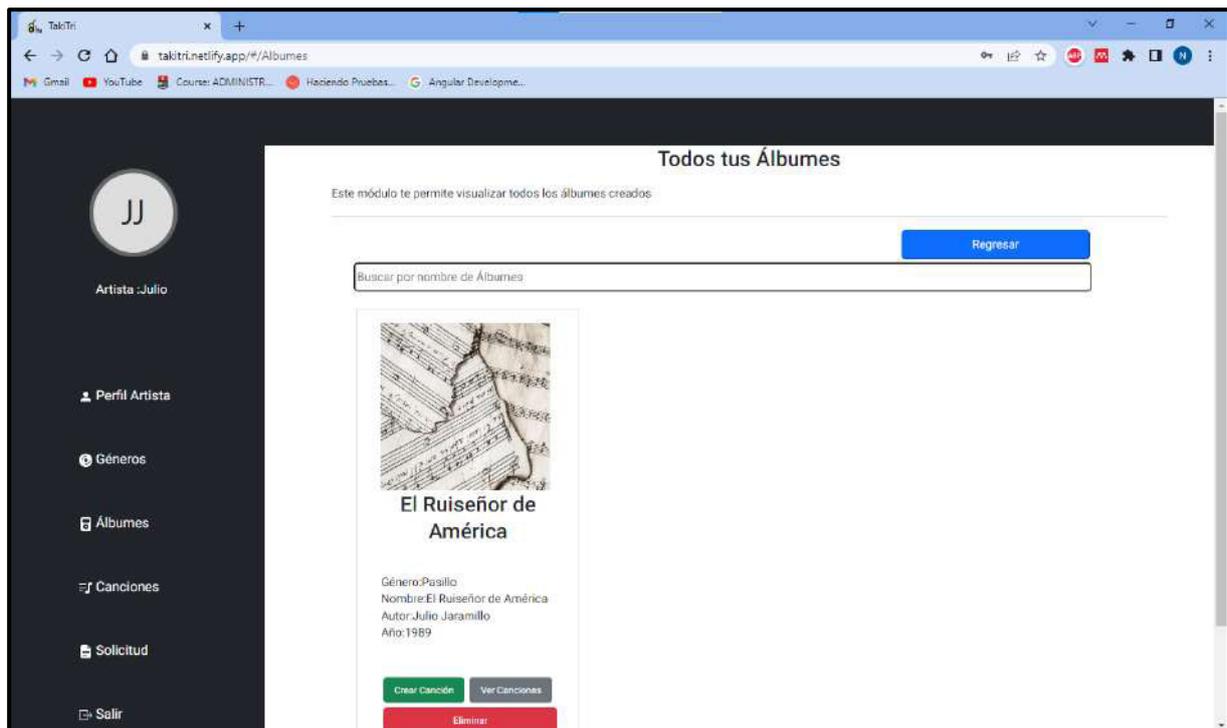


Fig.64: Página de Álbumes en el navegador Google Chrome.

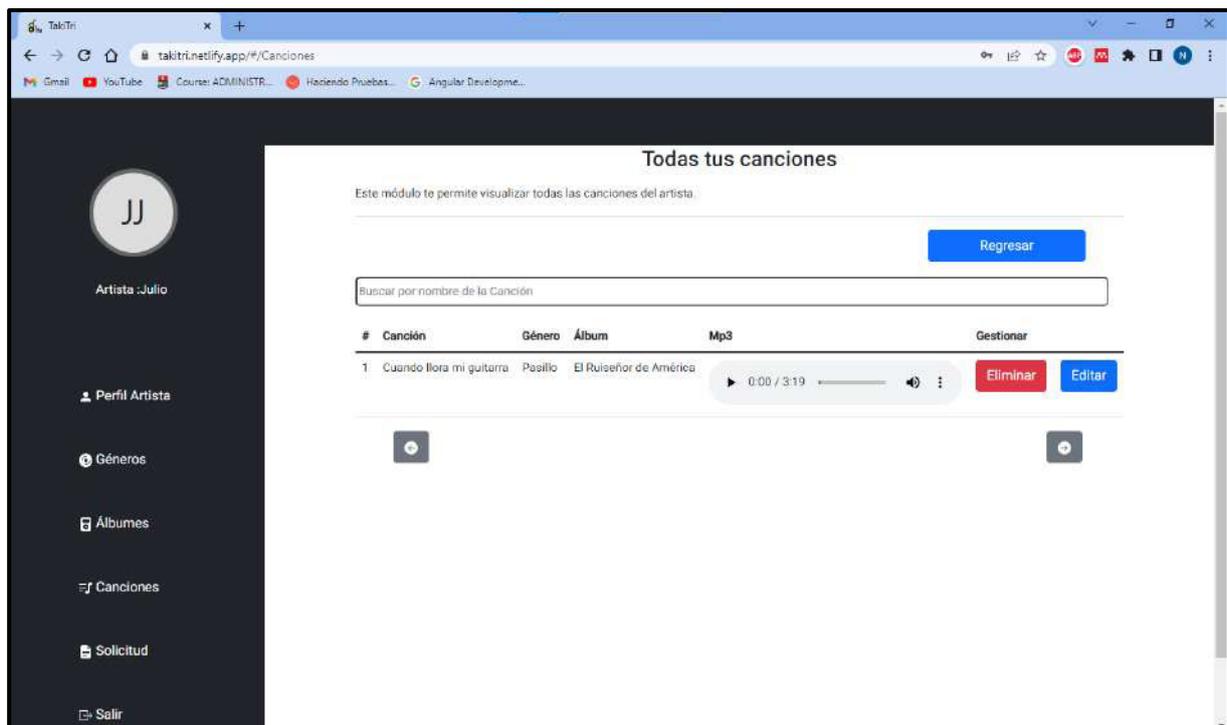


Fig.65: Página de Canciones en el navegador Google Chrome.

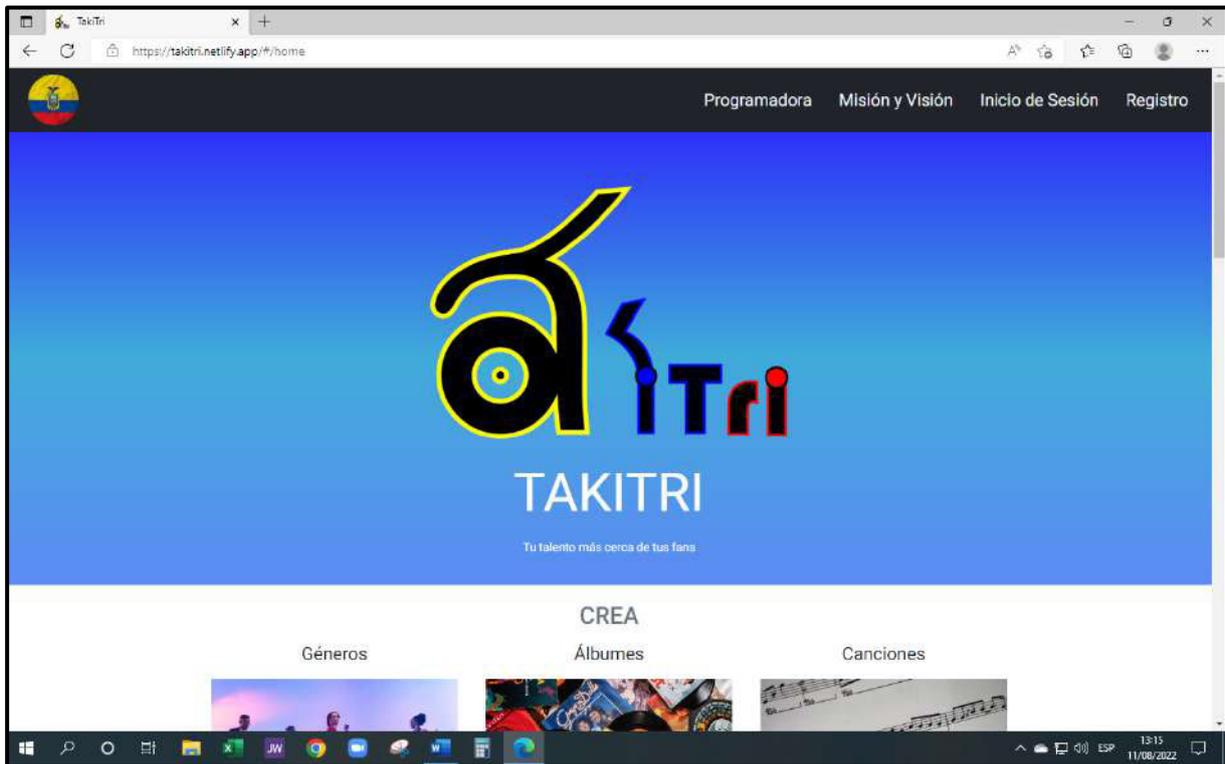


Fig.66: Página informativa vista en el navegador *Microsoft Edge*.

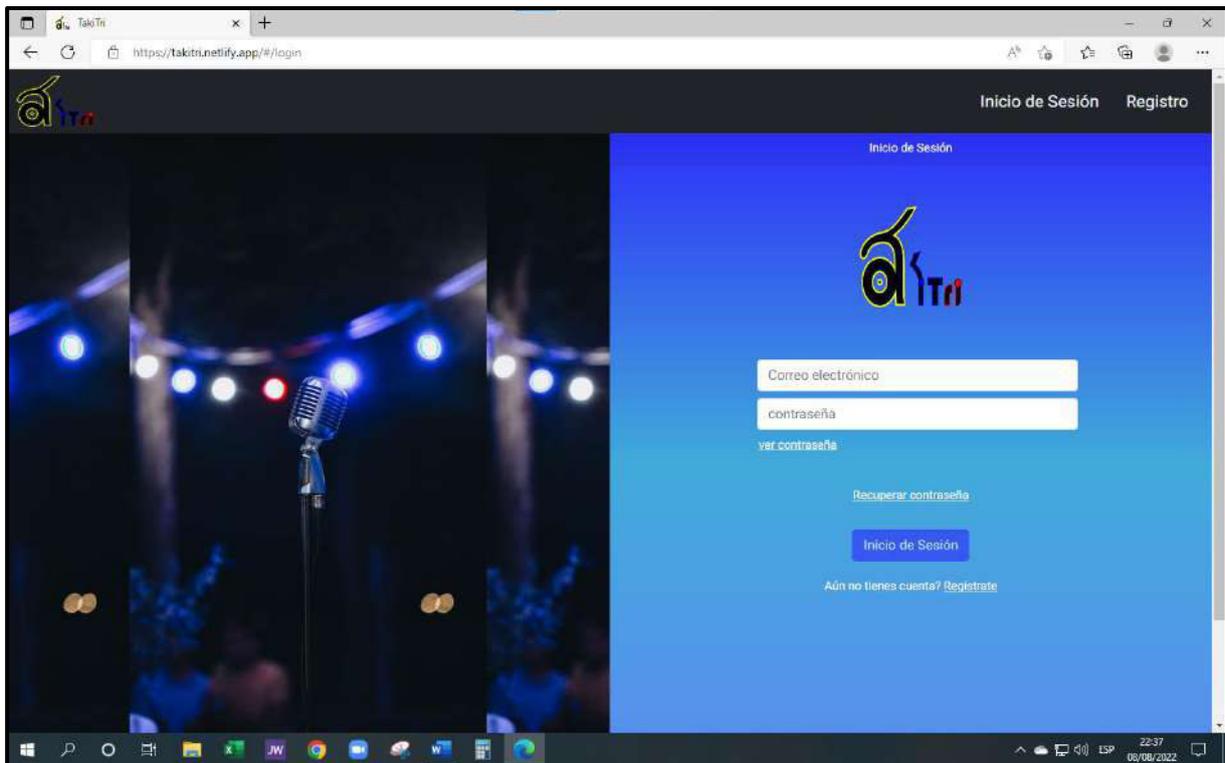


Fig.67: Página de Inicio de Sesión en el navegador *Microsoft Edge*.

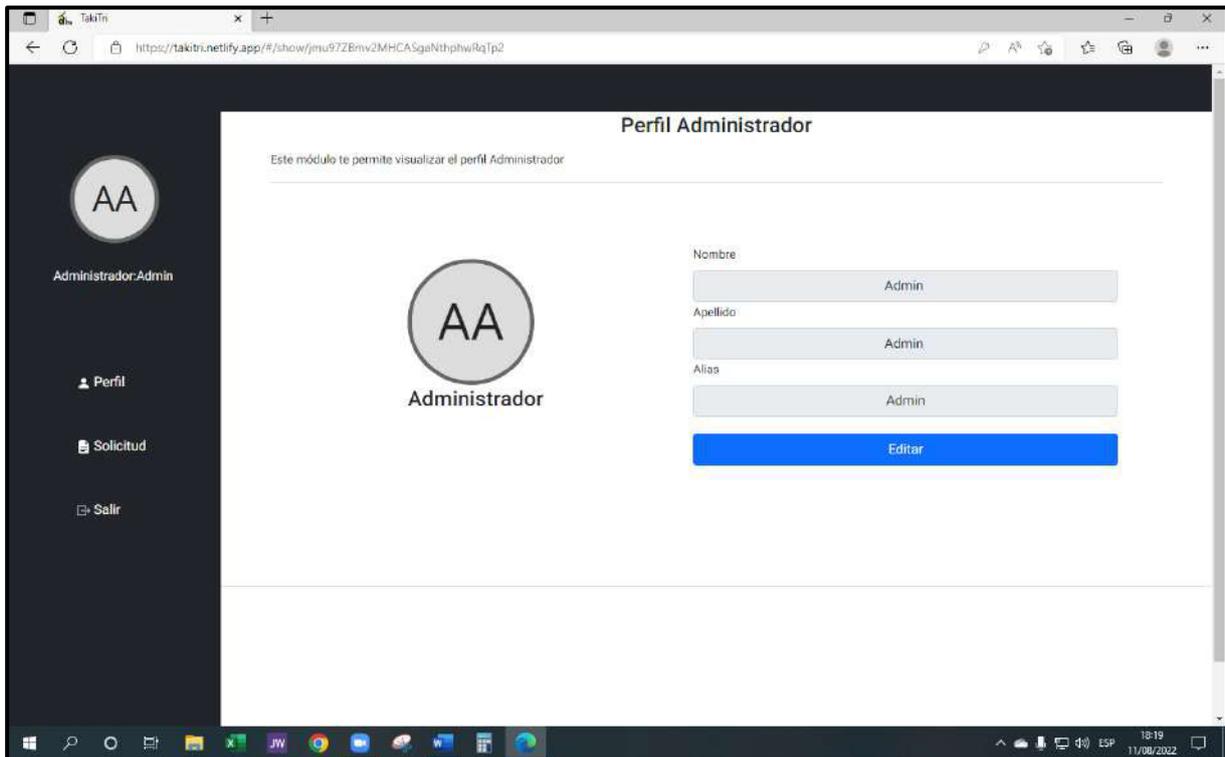


Fig.68: Página de Solicitudes en el navegador Microsoft Edge.

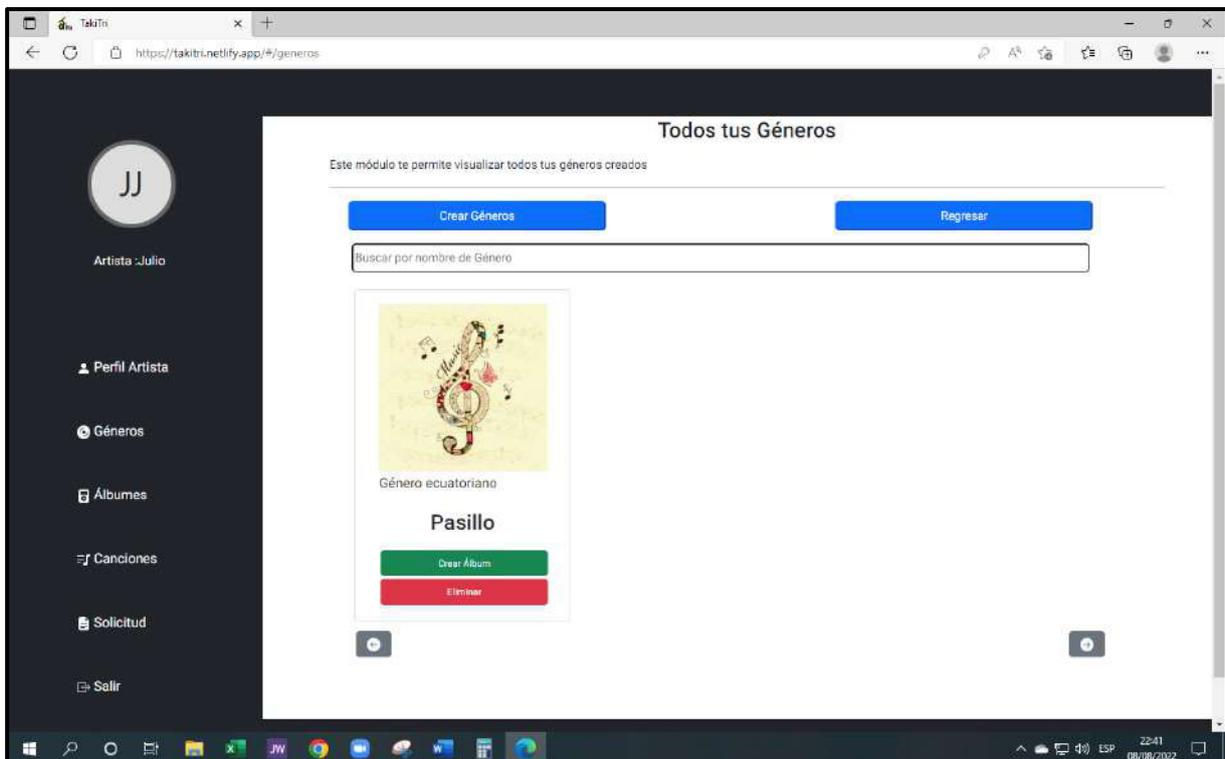


Fig.69: Página de Géneros en el navegador Microsoft Edge.

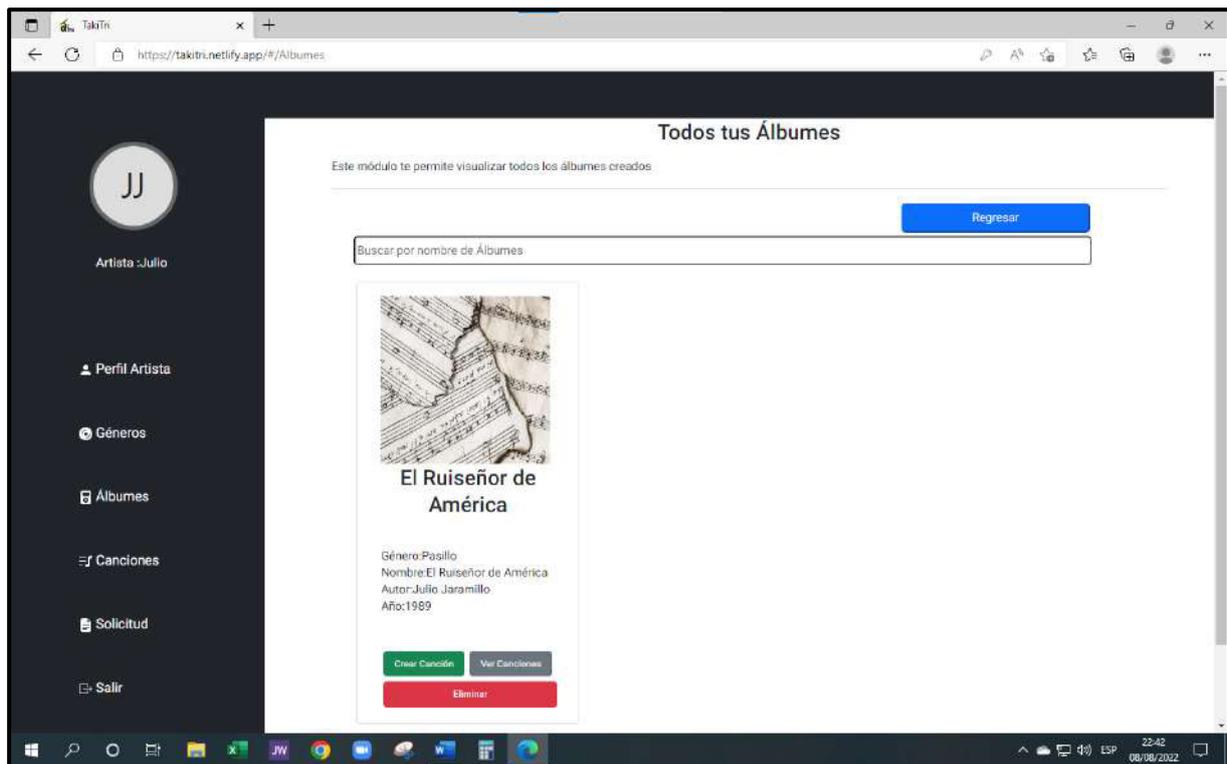


Fig.70: Página de Álbumes en el navegador Microsoft Edge.

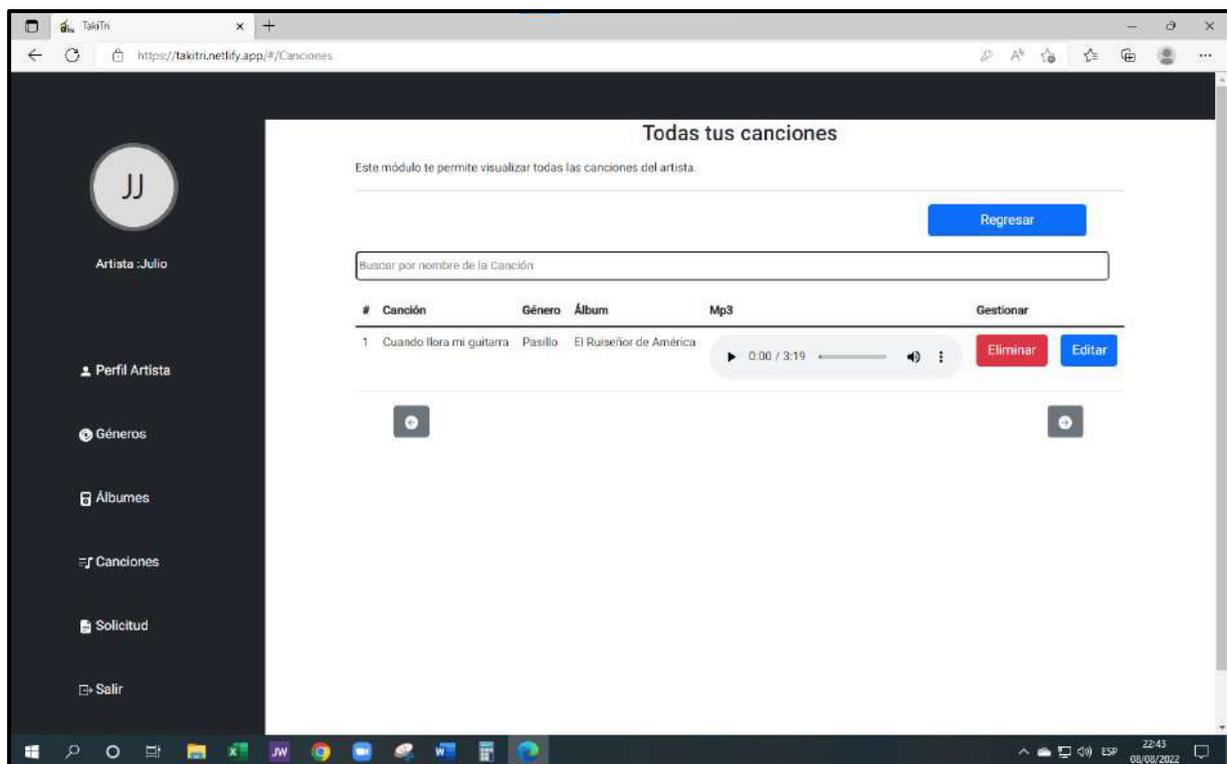


Fig.71: Página de Canciones en el navegador Microsoft Edge.

Prueba de aceptación

Se procede a mostrar las 10 pruebas de aceptación las cuales van desde la **TABLA XXII** hasta la **TABLA XXX**. Es importante mencionar que todas las pruebas describen el proceso que corresponde a las diferentes tareas asignadas de los usuarios del *frontend* para su correcta funcionalidad y verificación de cada una de ellas.

TABLA XXII: Prueba de aceptación 2 Consumir varios métodos para Iniciar sesión, cerrar sesión y recuperar contraseña.

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA002	Identificador de historia de Usuario: HU002
Nombre: Iniciar sesión, cerrar sesión y recuperar contraseña.	
Descripción: El usuario administrador y artista en el <i>frontend</i> puede consumir varios métodos para: <ul style="list-style-type: none">• Iniciar sesión• Cerrar sesión.• Recuperar contraseña.	
Pasos de ejecución: Para iniciar sesión usuario artista y administrador: <ul style="list-style-type: none">• Ubicar la URL del <i>frontend</i> en los navegadores.• En lado izquierdo en el menú superior dar clic en el botón “Iniciar Sesión”.• Poner el correo electrónico y contraseña dependiendo del rol de usuario.• Dar clic en “Ingresar”. Para cerrar sesión: <ul style="list-style-type: none">• Haber Iniciado sesión en el <i>frontend</i>.• En el menú lateral, en la parte final dar clic en salir. Para recuperar contraseña: <ul style="list-style-type: none">• Ubicar la URL del <i>frontend</i> en los navegadores.• Hacer clic en el menú Iniciar sesión.• Hacer clic en la parte inferior del formulario donde dice Recuperar contraseña.• Ubicar el correo electrónico en cual se haya registrado en el <i>frontend</i>.• Hacer clic en enviar.	

<ul style="list-style-type: none"> • Dirigirse a la cuenta de correo para revisar el correo enviado y hacer clic en el mismo. • Ubicar la nueva contraseña y enviar. • Dirigirse nuevamente al <i>frontend</i> y ubicarse en Iniciar sesión y poner la nueva contraseña.
<p>Resultado deseado:</p> <p>En TakiTri del <i>frontend</i> admite iniciar sesión y cerrar sesión y recuperar contraseña del usuario administrador y artista</p>
<p>Evaluación de la prueba:</p> <p>Se evidencia el resultado esperado. Conformidad del cliente 100%.</p>

TABLA XXIII: Prueba de aceptación 3 Consumir varios métodos para modificar perfil rol administrador.

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA003	Identificador de historia de Usuario: HU003
Nombre: Modificar perfil de usuario.	
<p>Descripción: El usuario administrador en el <i>frontend</i> puede consumir varios métodos para visualizar y editar su perfil por medio de los siguientes campos:</p> <ul style="list-style-type: none"> • Nombre. • Imagen. • Entre otros. 	
<p>Pasos de ejecución:</p> <p>Para editar perfil usuario administrador.</p> <ul style="list-style-type: none"> • Iniciar sesión en el sistema. • Hacer clic en el menú lateral en perfil. • Hacer clic en la parte inferior en editar. • Editar los campos permitidos • Hacer clic en editar. 	

<p>Resultado deseado:</p> <p>En TakiTri del <i>frontend</i> admite visualizar y editar el perfil del usuario administrador.</p>
<p>Evaluación de la prueba:</p> <p>Se evidencia el resultado esperado. Conformidad del cliente 100%.</p>

TABLA XXIV: Prueba de aceptación 4 Consumir varios métodos para gestionar solicitudes de artistas.

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA004	Identificador de historia de Usuario: HU004
Nombre: Gestionar solicitudes de artistas.	
<p>Descripción: El usuario administrador en el <i>frontend</i> puede consumir varios métodos para:</p> <ul style="list-style-type: none"> • Visualizar solicitudes de artistas. • Aprobar solicitudes de artistas. • Rechazar solicitudes de artistas. 	
<p>Pasos de ejecución:</p> <p>Para gestionar solicitudes por parte del usuario administrador.</p> <ul style="list-style-type: none"> • Iniciar sesión en el sistema como usuario administrador. • Visualizar las solicitudes por aprobar dependiendo si cumple con los requisitos hacer clic en aceptar o caso contrario en rechazar (escribir la razón de la denegación de solicitud). • Hacer clic en el menú de la tabla en aceptadas. • Visualizar todas las solicitudes aceptadas • Hacer clic en el menú de la tabla en rechazadas. • Visualizar todas las solicitudes rechaza y la razón de la decisión tomada por parte del administrador. 	

Resultado deseado:

En TakiTri del *frontend* admite aceptar solicitudes de los artistas, visualizar las solicitudes aceptadas y las rechazadas del usuario por parte del usuario administrador.

Evaluación de la prueba:

Se evidencia el resultado esperado.
Conformidad del cliente 100%.

TABLA XXV: Prueba de aceptación 5 Consumir varios métodos para Registrar el formulario.

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA005	Identificador de historia de Usuario: HU005
Nombre: Registrarse a través de un formulario.	
<p>Descripción: El usuario artista en el <i>frontend</i> puede consumir varios métodos para registrarse por medio llenando los siguientes campos:</p> <ul style="list-style-type: none"> • Nombre. • Alias. • Fecha de nacimiento. • Correo electrónico. 	
<p>Pasos de ejecución:</p> <p>Para registrar al usuario artista.</p> <ul style="list-style-type: none"> • Ubicar la URL del <i>frontend</i> en los navegadores. • Ubicar el nombre, alias, correo electrónico y fecha de nacimiento entre otros datos. • Hacer clic en registrarse. 	
Resultado deseado:	
En TakiTri del <i>frontend</i> admite registrar a usuarios con rol artista.	
Evaluación de la prueba:	
Se evidencia el resultado esperado. Conformidad del cliente 100%.	

TABLA XXVI: Prueba de aceptación 6 Consumir varios métodos para modificar perfil artista y enviar solicitud.

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA006	Identificador de historia de Usuario: HU006
Nombre: Modificar perfil de usuario y enviar solicitud	
<p>Descripción: El usuario artista en el <i>frontend</i> puede consumir varios métodos para modificar, visualizar su perfil además de enviar solicitudes por medio de los siguientes campos:</p> <p>Perfil artista</p> <ul style="list-style-type: none"> • Nombre. • Imagen. • Entre otros. <p>Solicitud</p> <ul style="list-style-type: none"> • Nombre artístico. • Número de cédula • Documento PDF. 	
<p>Pasos de ejecución:</p> <p>Para editar perfil usuario artista.</p> <ul style="list-style-type: none"> • Iniciar sesión en el <i>frontend</i>. • Hacer clic en el menú lateral en perfil. • Hacer clic en la parte inferior en editar. • Editar los campos permitidos • Hacer clic en editar. <p>Para enviar solicitud usuario artista.</p> <ul style="list-style-type: none"> • Iniciar sesión en el <i>frontend</i>. • Hacer clic en el menú latera en la parte inferior en solicitud. • Llenar un formulario con datos solidados y subir un pdf con la información requerida. • Hacer clic en enviar. 	

Resultado deseado:

En TakiTri del *frontend* admite enviar solicitudes del usuario artista.

Evaluación de la prueba:

Se evidencia el resultado esperado.

Conformidad del cliente 100%.

TABLA XXVII: Prueba de aceptación 7 Consumir varios métodos para gestionar géneros musicales.

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA007	Identificador de historia de Usuario: HU007
Nombre: Gestionar géneros musicales.	
<p>Descripción: El usuario artista en el <i>frontend</i> puede consumir varios métodos para:</p> <ul style="list-style-type: none"> • Crear géneros musicales. • Visualizar géneros musicales. • Eliminar géneros musicales. 	
<p>Pasos de ejecución:</p> <p>Para crear géneros musicales.</p> <ul style="list-style-type: none"> • Iniciar sesión en el sistema. • Haber enviado una solicitud y la misma ser aprobada por el administrador. • Hacer clic en géneros • Hacer clic en la parte superior en crear géneros. • Llenar los datos del formulario correctamente • Subir una imagen del género. • Hacer clic en enviar. <p>Para eliminar géneros musicales.</p> <ul style="list-style-type: none"> • Iniciar sesión en el sistema. • Haber enviado una solicitud y la misma ser aprobada por el administrador. 	

<ul style="list-style-type: none"> • Hacer clic en géneros en el menú lateral. • Seleccionar un género. • Hacer clic en eliminar géneros.
<p>Resultado deseado:</p> <p>En TakiTri del <i>frontend</i> admite crear, visualizar y eliminar los géneros musicales del usuario artista.</p>
<p>Evaluación de la prueba:</p> <p>Se evidencia el resultado esperado. Conformidad del cliente 100%.</p>

TABLA XXVIII: Prueba de aceptación 8 Consumir varios métodos para gestionar álbumes.

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA008	Identificador de historia de Usuario: HU008
Nombre: Gestionar álbumes.	
<p>Descripción: El usuario artista en el <i>frontend</i> puede consumir varios métodos para:</p> <ul style="list-style-type: none"> • Crear álbum. • Visualizar álbum • Eliminar álbum. 	
<p>Pasos de ejecución:</p> <p>Para crear álbum.</p> <ul style="list-style-type: none"> • Iniciar sesión en el sistema. • Haber enviado una solicitud y la misma ser aprobada por el administrador. • Hacer clic en géneros • Seleccionar un género. • Hacer clic en crear álbumes • Llenar los datos del formulario correctamente • Subir una imagen del álbum. 	

<ul style="list-style-type: none"> • Hacer clic en enviar. <p>Para eliminar álbumes.</p> <ul style="list-style-type: none"> • Iniciar sesión en el sistema. • Haber enviado una solicitud y la misma ser aprobada por el administrador. • Hacer clic en álbumes en el menú lateral. • Seleccionar un álbum ya creado con anterioridad. • Hacer clic en eliminar álbumes.
<p>Resultado deseado:</p> <p>En TakiTri del <i>frontend</i> admite crear, visualizar y eliminar los álbumes del usuario artista.</p>
<p>Evaluación de la prueba:</p> <p>Se evidencia el resultado esperado. Conformidad del cliente 100%.</p>

TABLA XXIX: Prueba de aceptación de aceptación 9 Consumir varios métodos para subir canciones.

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA009	Identificador de historia de Usuario: HU009
Nombre: Gestionar canciones.	
<p>Descripción: El usuario artista en el <i>frontend</i> puede consumir varios métodos para:</p> <ul style="list-style-type: none"> • Subir canción a un álbum. • Visualizar canciones. • Actualizar canción. • Eliminar canción. 	
<p>Pasos de ejecución:</p> <p>Para subir una canción.</p> <ul style="list-style-type: none"> • Iniciar sesión en el sistema. • Haber enviado una solicitud y la misma ser aprobada por el administrador. • Hacer clic en álbumes 	

- Seleccionar un álbum.
- Hacer clic en crear canción
- Llenar los datos del formulario correctamente
- Subir una canción del artista.
- Hacer clic en enviar.

Para editar canción.

- Iniciar sesión en el sistema.
- Haber enviado una solicitud y la misma ser aprobada por el administrador.
- Hacer clic en canciones en el menú lateral o seleccionado un álbum en ver canciones.
- Seleccionar una canción ya creado con anterioridad.
- Hacer clic en editar.
- Visualizar la información de la canción seleccionado y modificar su contenido en los campos permitidos.
- Hacer clic en enviar.

Para eliminar canción.

- Iniciar sesión en el sistema.
- Haber enviado una solicitud y la misma ser aprobada por el administrador.
- Hacer clic en canciones en el menú lateral o seleccionado un álbum en ver canciones.
- Seleccionar una canción ya creado con anterioridad.
- Hacer clic en eliminar.

Resultado deseado:

En TakiTri del *frontend* admite subir, visualizar, editar y eliminar las canciones del usuario artista.

Evaluación de la prueba:

Se evidencia el resultado esperado.
Conformidad del cliente 100%.

TABLA XXX: Prueba de aceptación 10 Consumir varios para Buscar canciones.

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA0010	Identificador de historia de Usuario: HU0010
Nombre: Buscador de canciones.	
Descripción: El usuario artista en el <i>frontend</i> tiene la posibilidad de buscar canciones.	
<p>Pasos de ejecución:</p> <p>Para buscar canciones del artista.</p> <ul style="list-style-type: none"> • Iniciar sesión en el sistema. • Haber enviado una solicitud y la misma ser aprobada por el administrador. • Hacer clic en el menú lateral en canciones. • Ubicar el nombre de la canción en el buscador que está en la parte superior. 	
<p>Resultado deseado:</p> <p>En TakiTri del <i>frontend</i> admite buscar las canciones del usuario artista.</p>	
<p>Evaluación de la prueba:</p> <p>Se evidencia el resultado esperado. Conformidad del cliente 100%.</p>	

ANEXO III

A continuación, para visualizar el Manual de Usuario del *frontend* se debe digitar la siguiente URL:

<https://youtu.be/KX-hZYJr1JY>

En donde se explica de forma clara y sencilla las diversas funcionalidades del *frontend*, así como cada uno de los perfiles que forman parte de este componente.

ANEXO IV

A continuación, se presenta las credenciales de acceso del *frontend*, además del repositorio de GitHub, donde se encuentra todo el código de TakiTri, además de los pasos a seguir para su instalación en el apartado del README.

Credenciales para el acceso del *frontend*.

Para ingresar al *frontend* ya en producción, se ingresa mediante la URL:

<https://takitri.netlify.app/#/home>

Credenciales del perfil administrador:

- **Correo del administrador:*******
- **Contraseña:** *****

Credenciales para el perfil artista:

- **Correo de artista:** *****
- **Contraseña:** *****

Repositorio del *frontend*

El proyecto se encuentra en un repositorio de *GitHub*, que se accede mediante la siguiente URL:

https://github.com/Noe19/Noemi_taki_tri.git