



**UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS**

CREACIÓN DE UN PROTOTIPO DE APLICACIÓN MÓVIL EN DISPOSITIVOS ANDROID PARA LA CONTRATACIÓN Y PROMOCIÓN DE MÚSICOS

**Gustavo Alejandro Realpe Fresneda
David Fernando Cabarcas Bolaños**

Universidad Distrital Francisco José de Caldas
Especialización en Ingeniería de Software - Grupo 2
Bogotá, Colombia
Año 2018



**UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS**

CREACIÓN DE UN PROTOTIPO DE APLICACIÓN MÓVIL EN DISPOSITIVOS ANDROID PARA LA CONTRATACIÓN Y PROMOCIÓN DE MÚSICOS

**Gustavo Alejandro Realpe Fresneda
David Fernando Cabarcas Bolaños**

**Director: Giovanny Tarazona
Revisor: Julio Baron Velandia**

Universidad Distrital Francisco José de Caldas
Especialización en Ingeniería de Software - Grupo 2
Bogotá, Colombia
Año 2018

Contenido

INTRODUCCIÓN

La realización de este proyecto supone un fondo social en el que se busca facilitar y mejorar la calidad de vida de las personas dedicadas a la prestación de servicios de entretenimiento que no tienen un conocimiento administrativo o de mercadeo adecuado para promocionar esta labor. Lo anterior toma mayor valor teniendo en cuenta la amplia oferta de programas educativos en programas universitario de música y arte que están produciendo gran número de egresados, que no encontrarán una demanda laboral favorable y se perderán en un mercado hostil que probablemente los llevará a desempeñarse en otras labores contrarias a su vocación. Por otra parte la informalidad que existe en el proceso de contratación de esta labor artística ha generado todo tipo de formas de contacto entre el oferente y el demandante y es por esto que el mercado es tan difuso e irregular. La centralización de la oferta permite mejorar la calidad y normaliza el mercado logrando así el mejor valor para los implicados en el proceso[?][?].

Para suplir dicha necesidad nos aventuramos en el proceso de desarrollo de una aplicación móvil innovadora enmarcada en los principios de desarrollo ágil aplicando los lineamientos de buenas prácticas. Lo anterior para fundamentar los conocimientos adquiridos a través de la experiencia en ingeniería de software.

En este proceso ejecutamos el proceso investigativo que permite inferir la mejor opción para cubrir esta necesidad; realizando el análisis del proceso actual de selección de un servicio y contratación del mismo. Se realizará el modelamiento y diseño necesario inicial para la generación de un producto sólido y estable desde sus primeras versiones limitando la posibilidad de falla a casos imprevistos que se susciten solo a través del uso de la herramienta y no a la falta de planeación de las mismas.

Parte I

CONTEXTUALIZACIÓN DE LA INVESTIGACIÓN

Capítulo 1

DESCRIPCIÓN DE LA INVESTIGACIÓN

1.1. Planteamiento del problema

La situación económica del artista colombiano, en general, no ha sido nunca buena. Por supuesto, hay excepciones. Algunos pintores, escultores, músicos y escritores ganan mucho dinero. Son un puñado. En la pirámide hacia abajo un conglomerado sobrevive. Pero en la base una muchedumbre pasa las duras y las maduras. Si habláramos en términos de desempleo, la tasa artística podría alcanzar el 80 por ciento. Obviamente el desempleo entre los creadores de arte es relativo. Un creador podría ejecutar excelencias en un sótano oscuro en circunstancias de supervivencia dramáticas. De cualquier forma, la tendencia del estatus del artista en estos tiempos se inclina hacia la comodidad, el progreso técnico y el éxito económico. Por tanto, en este país, donde tras 23 años de lucha por obtener seguridad social inútilmente para el artista, no sorprende que dos grandes figuras se lancen a la aventura de la empresa artística, solo para abrirle posibilidades a la tasa de desempleados [?].

Muchos de estos artistas se han hecho a pulso, son muy talentosos pero no han podido concretar sus estudios universitarios en este arte. Comenzando por el costo de la carrera, que a diferencia del sector público (al que no todos pueden ingresar), en el privado, el precio por semestre oscila entre \$3 millones y \$11 millones; sumado a eso, los gastos de fotocopias, transporte, e instrumentos, que para cumplir medianamente con los estándares mínimos de calidad representan una cantidad de dinero considerable [?].

La informalidad laboral es un concepto que va más allá de la creencia de que solo las personas que trabajan en la calle o independientes son consideradas informales. Este término también abarca a los trabajadores a los que no se les ha legalizado su labor o a los que de alguna manera se les ha trasgredido alguno de los requisitos establecidos por la Organización Internacional del Trabajo (OIT) [?].

Los músicos del rebusque intentan mantenerse —y mantener a sus familias— con lo que diariamente producen en las calles: pueden recoger entre 20 mil y 150 mil pesos en un solo día, todo depende del flujo de peatones y de la fecha, porque si pagaron la quincena la gente es más generosa. Muchos de estos intérpretes no pagan ni salud ni pensión. Según una encuesta realizada por el DANE, publicada el pasado 8 de abril, entre noviembre de 2015 y febrero de este año el empleo informal en Bogotá fue del 47,1 %. La cifra bajó tan solo un 1 % con respecto al mismo trimestre del año pasado [?].

“basados en datos del Banco Interamericano de Desarrollo (BID), que la industria cultural aporta al Producto Interno Bruto (PIB) de Colombia un 3 %, pero señala que la inversión que tiene el Estado para la misma es de un 0,16 %, por lo que le parece paradójico que en Colombia no haya oportunidades suficientes para que la gente estudie o trabaje en ese campo”: David García ex-director general de la Orquesta Filarmónica de Bogotá. [?].

“Estamos comprometidos con el impulso a la economía naranja para que nuestros actores, artistas, productores, **músicos**, diseñadores, publicistas, joyeros, dramaturgos, fotógrafos y animadores digitales conquisten mercados, mejoren sus ingresos, emprendan con éxito”, dijo el presidente de Colombia Iván Duque, cuando tomó posesión del cargo [?]. Aún así, entre los retos que registra este modelo, según indica Duque, está en el de “retener, atraer, capturar y reproducir el talento de un segmento de la población, que por lo general se encuentra subvalorado socialmente y pobremente remunerado económicamente” [?], y acá es donde se quiere atacar el problema.

A través de la innovación facilitar la forma en que se ofrecen servicios ha sido una de las mayores utilidades que se le ha dado a internet, el problema que queremos solventar es especializar la oferta de un servicio que hasta el momento no ha sido visualizado. Aplicar todo el conocimiento de ingeniería de software para producir un producto con código limpio, escalable, probable y todas las características que lo identifiquen como de alta calidad. Planeamos realizar todo el ciclo de vida de un desarrollo aplicando metodologías y tecnologías de última generación que nos conduzcan a afianzar los conocimientos del arte tecnológico implicado en el proceso de producción de software.

La idea de generar un prototipo de aplicación móvil para centralizar la oferta y demanda de servicios musicales supone el facilitar el ambiente en el que se genera dicha interacción. Por lo anterior proponemos agregar a la aplicación un módulo de comunicación que permita el contacto directo entre oferente y demandante además de la posibilidad de historial de servicios con calificaciones que certifiquen la calidad del servicio prestado igualmente la calificación de los demandantes como personas confiables y respetables.

1.2. Formulación del problema

¿Cómo brindarles a los músicos un prototipo de una herramienta digital implementada con las mejores prácticas de desarrollo de software que facilite la búsqueda, dar a conocer y contratación de músicos informales o en fase de descubrimiento?

1.3. Sistematización del problema

- ¿Cómo centralizar la oferta y demanda de servicios musicales para facilitar el ambiente en el que se genera dicha interacción?
- ¿Cuál es la arquitectura que más se ajusta para solucionar este tipo de problema?
- ¿Cómo mejorará los ingresos de los músicos asociados a la plataforma?
- ¿Cuál es la información más relevante que un usuario tiene en cuenta para contratar a un músico?

1.4. Objetivo general

Generar un prototipo de aplicación móvil con las mejores prácticas de desarrollo que permita la promoción de servicios musicales por medio de vídeos, comentarios, calificaciones, para centralizar la oferta y demanda de servicios de entretenimiento musical.

1.5. Objetivos específicos

- Modelar una aplicación móvil que permita a los artistas informales promocionarse por medio de contenido multimedia y presente un protocolo claro para la contratación de sus servicios.
- Producir una herramienta tecnológica de fácil acceso para el público en general que permita relaciones confiables y de valor por medio de procesos de trazabilidad clara.

1.6. Justificación del proyecto

Se necesitan herramientas que puedan unir y comunicar todos los actores del problema, que son los artistas, el público, las entidades públicas y privadas, en la medida que los artistas necesitan del público, los escenarios necesitan eventos y artistas, y los sectores público y privado necesitan garantizar a la comunidad el acceso y la difusión a la cultura a través de una actitud de compromiso social que conlleve al objetivo del mejoramiento de la calidad de vida de las personas[?].

A los músicos se les recomienda tener un manager para establecer las relaciones comerciales, por que su formación no es de administrador, pero se han abierto un mundo a través de las alianzas que en muchos casos los proyectan en una labor más administrativa o artística. En casos donde los artistas estén en sus primeras fases, no va a ser necesario que ellos adquieran los servicios de un manager si a través de una plataforma digital enfocada hacia ellos, puedan promocionar sus servicios, mostrar sus talentos y tienen la posibilidad de ser contratados directamente[?].

En el aspecto laboral, la informalidad y el desempleo en este sector son altos, y la oferta laboral presentada por cuenta de las industrias culturales es baja, y hay pocas organizaciones que demanden los servicios de los artistas y en muchos casos no hay suficiente información sobre los artistas, sobre la calidad y los servicios que prestan; y la contratación muchas veces se hace por recomendaciones personales o contrataciones previas, lo que lleva a pérdidas de oportunidades de muchos artistas que por no ser conocidos o referenciados no llegan a ser contratados[?].

Identificada la necesidad de promoción de artistas, para evitar los intermediarios y que tengas más oportunidades laborales, este proyecto se orienta al desarrollo de un prototipo de una aplicación móvil, en donde se van a servir de puente en los usuarios y los artistas.

1.7. Hipótesis

Una aplicación móvil que permita centralizar la información de los artistas, y que pueda ofrecerla a los clientes interesados en sus servicios, facilitando la comunicación, calificación y contratación de los mismos.

1.8. Marco Referencial

1.8.1. Marco Teórico

Antecedentes

Tu toque, página web donde los músicos pueden ofrecer sus servicios con precios, formatos, géneros y estilos definidos; y los clientes pueden buscar, filtrar, comparar y contratar al músico ideal. Esta plataforma permite, con base en los criterios de búsqueda del usuario, buscar por artista, ubicación, género musical y además filtrar por precios y calificaciones. Los perfiles de músicos ofrecen diferentes servicios y formatos musicales, lo cual brinda al cliente un gran número de opciones para elegir. Asimismo, el cliente cuenta con retroalimentación de clientes anteriores para formarse una mejor idea del servicio ofrecido por el músico. Con el fin de dar tranquilidad a músicos y clientes, el pago se hace por anticipado, pero el músico recibe el dinero después de haber prestado el servicio, de manera que tuToque.co sirve como garante entre las partes [?] [?].

JamsOut, Sitio web que funciona en Ecuador, es un buscador de artistas, permite a cantantes e instrumentistas colgar su información, más enlaces de sus trabajos en el portal. La información debe constar de fotos y videos de sus conciertos, links de sus canciones en Spotify, iTunes o Deezer. También debe colocar una reseña artística de 20 líneas, en la cual se hable de su trayectoria, escenarios donde ha cantado y una descripción del género musical en el que se desenvuelve.

Inkasha, es una red social en línea que busca promover el talento y la interacción entre bandas de Colombia a través de la creación de tres diferentes perfiles donde ellos pueden mostrar su música, descubrir trabajo, conectarse con otros músicos, y encontrar venues donde tocar. Estos perfiles se dividen en músicos (solistas que busquen promover su música o que estén buscando una banda a la cual pertenecer), bandas (que busquen participar en convocatorias, mostrar su música o encontrar integrantes para sus agrupaciones) y dueños de venues (auditorios, bares, cafés, hoteles, universidades, etc. donde busquen a músicos para presentarse en sus espacios). Y los usuarios interesados pueden ingresar y mirar la información del artista y elegir si quieren sus servicios[?].

1.8.2. Marco Conceptual

Android

Sistema operativo basado en Linux, fue desarrollado para dispositivos móviles con pantalla táctil, al día de hoy es el sistema operativo más usado del mundo por encima de Ios de Apple. Los dispositivos con este sistema operativo se puede programar en Java o en Kotlin.

NoSql

Son muchas las aplicaciones web que utilizan algún tipo de bases de datos para funcionar. Hasta ahora estábamos acostumbrados a utilizar bases de datos SQL como son MySQL, Oracle o MS SQL, pero desde hace ya algún tiempo han aparecido otras que reciben el nombre de NoSQL (Not only SQL – No sólo SQL) y que han llegado con la intención de hacer frente a las bases relacionales utilizadas por la mayoría de los usuarios [?].

Ventajas:

Esta forma de almacenar la información ofrece ciertas ventajas sobre los modelos relacionales. Entre las ventajas más significativas podemos destacar:

- Se ejecutan en máquinas con pocos recursos: Estos sistemas, a diferencia de los sistemas basados en SQL, no requieren de apenas computación, por lo que se pueden montar en máquinas de un coste más reducido.
- Escalabilidad horizontal: Para mejorar el rendimiento de estos sistemas simplemente se consigue añadiendo más nodos, con la única operación de indicar al sistema cuáles son los nodos que están disponibles.
- Pueden manejar gran cantidad de datos: Esto es debido a que utiliza una estructura distribuida, en muchos casos mediante tablas Hash.
- No genera cuellos de botella: El principal problema de los sistemas SQL es que necesitan transcribir cada sentencia para poder ser ejecutada, y cada sentencia compleja requiere además de un nivel de ejecución aún más complejo, lo que constituye un punto de entrada en común, que ante muchas peticiones puede ralentizar el sistema.

Arquitectura Limpia

Robert Martin, más conocido como Tío Bob (Uncle Bob en inglés), plantea que la arquitectura de un software debe gritar cuál es el uso de dicho software. Así como al ver planos de edificios, si uno ve que el edificio posee un baño, cocina, comedor, dormitorios, la arquitectura grita que es una casa familiar. De esta manera, al visualizar la arquitectura de un software, se debería notar cual es el uso del mismo. La arquitectura no debería indicar cuál es el framework o herramientas usadas para su construcción.

Volviendo al arquitecto de edificios, el mismo se asegura que un edificio cumpla con los casos de uso antes de decidir qué materiales se usarían para su construcción. En el caso del desarrollo de software, complementar esto, asegura que el producto satisfaga los casos de uso y retrase la decisión sobre los frameworks o herramientas a usar. Incluso, una buena arquitectura basada en casos de usos permite que sea fácil cambiar la herramienta que se va a usar [?].

La propuesta del Tío Bob sobre la arquitectura limpia impone las siguientes restricciones:

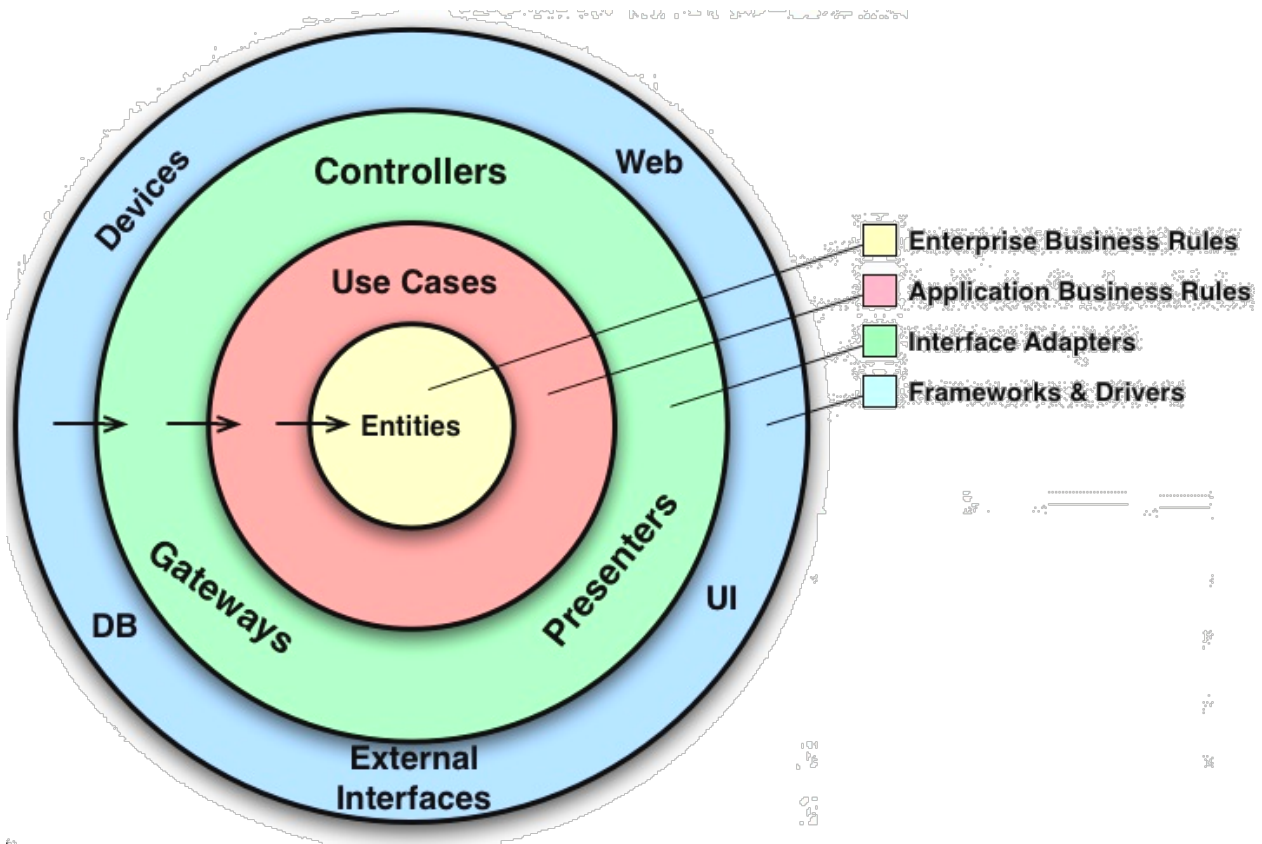


Figura 1-1: Arquitectura Limpia.

- **Independiente de frameworks:** la arquitectura no debe atarse a las restricciones del framework. Esto permite considerar al framework como una herramienta.
- **Testeable:** las reglas de negocios deberían poder probarse independientemente de la UI, base de datos u otras herramientas.
- **Independiente de la UI:** la interfaz de usuario se debería poder cambiar fácilmente sin necesidad de cambiar la lógica de negocio.
- **Independiente de la base de datos:** las reglas de negocios no deberían estar ligadas a la base de datos o herramienta de persistencia.

MVP

Es una derivación del patrón arquitectónico modelo–vista–controlador (MVC), y es utilizado mayoritariamente para construir interfaces de usuario

En MVP el presentador asume la funcionalidad del “intermediario”. En MVP, toda lógica de presentación es colocada al presentador [?].

MVP es un patrón arquitectónico de interfaz de usuario diseñada para facilitar pruebas de unidad automatizada y mejorar la separación de inquietudes en lógica de presentación:

- El modelo es una interfaz que define los datos que se mostrarán o no actuado en la interfaz de usuario.
- El presentador actúa sobre el modelo y la vista. Recupera datos de los repositorios (el modelo), y los formatea para mostrarlos en la vista.
- La vista es una interfaz pasiva que exhibe datos (el modelo) y órdenes de usuario de las rutas (eventos) al presentador para actuar sobre los datos.

SISTEMAS DE RECOMENDACIÓN

Es un sistema inteligente que proporciona a los usuario sugerencias personalizadas sobre un determinado tema. Estudian el perfil del cliente, sus búsquedas anteriores, sus compras, perfiles, páginas visitadas, recopilan toda esta información e intentan hacer sugerencias que son de interés del usuario.

Uno de los sistemas de recomendación famosos son, Amazon e Youtube, la primera utiliza la información de las compras de los usuarios, la navegación y búsqueda de los productos para sugerir posibles productos de interés. Youtube registra la búsqueda de vídeos y la visualización de vídeos para mostrar sugerencias de vídeos que le podrían gustar. [?]

Los sistemas de clasificación se pueden dividir en 4 tipos:

1. **Filtrado basado en Contenido** Las recomendaciones se basan en el conocimiento que se tiene sobre los items que el usuario ha valorado (ya sea de forma implícita o explícita), y se le recomendarán items similares que le puedan gustar o interesar. Un ejemplo de es Youtube.
2. **Filtrado Demográfico:** Estas recomendaciones se realizan en función de las características de los usuarios (edad, sexo, situación geográfica, profesión, etc).
3. **Filtrado Colaborativo:** Consiste en ver que usuarios son similares al usuario activo (o usuario al que hay que realizarle las recomendaciones) y a continuación, recomendar aquellos items que no han sido votados por el usuario activo y que han resultado bien valorados por los usuarios similares. Un ejemplo de es Filmaffinity.
4. **Filtrado Híbrido:** Mezclan alguno de los tres filtrados mencionados anteriormente para realizar recomendaciones e incluso lo combinan con alguna otra técnica de inteligencia artificial como pueda ser la lógica borrosa o la computación evolutiva. Un ejemplo es Amazon.



Figura 1-2: Tipos de sistema de recomendación. [?]

MICROSERVICIOS

Se define un microservicio es una pequeña aplicación que puede ser desplegada, probada y escalada de forma independiente y que tiene una única responsabilidad[?]. Hablamos de única responsabilidad en su definición básica, que es que solo se tenga que cambiar por sola una cosa [?].

¿Qué se debe tener en cuenta?

Se espera que un microservicios no tengo muchas líneas de código y puedas entrar todo su poder a sólo una funcionalidad. Normalmente cuando se empieza en un proyecto, se inicia con una aplicación monolítica, que para un inicio están bien, pero a medida que el proyecto va creciendo se vuelve más complicado su mantenibilidad, su escalabilidad, sus pruebas debido a que todos los proyectos los servicios están unidos, haciendo que los despliegues se vuelvan complicados y muy grandes. Al tener una aplicación monolítica, hace que un fallo en uno de los servicios puede tumbar toda nuestra aplicación, una cosa que no deseamos. Y también si queremos hacer un pequeño cambio en nuestro código significa que tenemos que desplegar toda la aplicación, no se puede desplegar sobre el pequeño cambio que hicimos. Estos son unos de los problemas que intenta atacar la arquitectura por microservicios, y por lo cuál en los últimos tiempos se han vuelto tan utilizada, haciendo servicios compactos e independientes hacemos que se pueda tomar ventaja de todas las tecnologías que hay en el mercado pudiendo a cada uno de los servicios especializarlo en la tecnología que más le convenga, estos servicios son especializados, o sea que sólo tienen una responsabilidad lo cual los hace fácil de escalar de mantener y de probar. Un módulo común de esta arquitectura es que tenga la lógica de negocio asociado con la base de datos la base de datos Y es única para ese servicio, esto implica que si queremos hacer un cambio en ese servicio sólo tenemos que desplegar ese servicio sin afectar los demás, asimismo, si un servicio falla todos los otros servicios de nuestra aplicación pueden seguir funcionando.

¿Es buena idea iniciar de una con microservicios?

Cuando se decide implementar una arquitectura por microservicios debe tener muy claro la lógica del negocio, porque hay que saber muy bien cómo se va a dividir el proyecto en los diferentes servicios, en cada uno de los servicios para así no tener problemas más adelante, la mayoría de empresas que han implementado microservicios, lo primero que han tenido es una aplicación monolítica, una aplicación con todos sus servicios integrados y a medida que van creciendo y ya con la lógica de negocio clara, definen los servicios que necesitan y empiezan a dividir su código monolítico en servicios más pequeños y especializados [?].

¿Es compleja la migración?

La respuesta concisa es que si, al tener muchos microservicios empezamos a tener necesidades que no teníamos antes con las aplicaciones monolíticas, tenemos que tener en cuenta que pueden ser muchos servicios que se van a estar comunicando entre ellos y hacia el exterior, aquí es donde nos empieza a apoyar todas las herramientas de despliegue continuo y computación en la nube, al tener muchos servicios nos encontramos con el inconveniente de tener que hacer muchos despliegues y muchas configuraciones en ambientes separados lo cual sin herramientas que nos ayuden a esta tarea sería muy complicado de mantener, adicionalmente pensando en la escalabilidad de un sistema nos damos cuenta de que esos servicios se pueden duplicar, triplicar, ..., dependiendo de la carga dependiendo lo que de las configuraciones que tengamos, lo cual hace que no podemos tener direcciones o apuntador servicios estáticos, tenemos que tener herramientas o servicios que nos permitan dinámicamente encontrar todos los microservicios de nuestra aplicación, con esto queremos decir que una aplicación de microservicios no es sólo dividir servicios y ya, detrás de eso tenemos lógicas esenciales y herramientas para que esto pueda funcionar la forma correcta y que no se nos vuelva un problema al final. Para esto vamos a utilizar toda la parte de integración continua y adicionalmente servicios de Gateway que nos permitan encontrar y distribuir nuestros servicios [?][?].

1.8.3. Marco Legal

Papel del estado y las instituciones públicas

Al Estado le corresponde:

Promover y fomentar el acceso a la cultura de todos los colombianos en igualdad de oportunidades, por medio de la educación permanente y la enseñanza científica, técnica, artística y profesional; Promover la investigación la ciencia, el desarrollo y la difusión de los valores culturales de la nación (Art 70. Constitución Política)

Crear incentivos para personas e instituciones que desarrollen y fomenten la ciencia y la tecnología y las demás manifestaciones culturales y ofrecer estímulos a quienes ejerzan estas actividades.

(Artículo 71. Constitución Política).

Impulsar y estimular los procesos, proyectos y actividades culturales en un marco de reconocimiento y respeto por la diversidad. (Artículo 1-3, ley 397 de 1997)

Abstenerse de ejercer censura sobre la forma y el contenido ideológico y artístico de las realizaciones culturales y proyectos culturales (Art, 1-4, ley 397 de 1997)

Promover la interacción de la cultura nacional con la cultura universal (Art, 1-12, ley 397 de 1997)

Garantizar el acceso de los colombianos a las manifestaciones, bienes y servicios culturales. (Art, 1-13, ley 397 de 1997) [?]

1.9. Aspectos metodológicos

1.9.1. Tipo de estudio

El tipo de investigación utilizado en la generación de este prototipo será proyectiva/interactiva[?]. lo anterior teniendo en cuenta que la idea central del proyecto es proponer una nueva forma de realizar un proceso de contratación que modifique la que en este momento se utiliza de manera común, mejorandola y proyectándola a posiblemente impulsar nuevos talentos y apoyarlos en procesos administrativos y de negocios a los que no están acostumbrados comportándose como un manager para artistas informales que de pronto no cuenten con los recursos necesarios para adquirir dichos servicios.

1.9.2. Método de investigación

Para lograr el fin deseado utilizaremos apartes de cada uno de los niveles de la investigación holística de la siguiente manera:

Perceptual: Exploraremos las opciones utilizadas en la actualidad para lograr la contratación de un servicio de entretenimiento; describiéndolo de manera que encontremos sus falencias y oportunidades de mejora.

Aprehensivo: Compararemos las formas en las que se realiza el proceso de selección de artistas informales y analizaremos la que produzca mayor impacto para extraer de hay los puntos fuertes que deben incluirse en nuestra propuesta.

Comprensivo: Propondremos una nueva opción que cubra las necesidades de veladas en los pasos anteriores.

Integrativo: Apuntaremos a modificar la forma en que se realiza el proceso de selección al momento de elegir un artista informal para una presentación

1.9.3. Fuentes y técnicas para la recolección de la información

La información primaria se obtendrá de la aplicación de una encuesta que determine la forma de contratación de servicios artísticos en una población controlada mayor de edad que esté en la capacidad de realizar dicho proceso, se generan preguntas cerradas que limitan las posibilidades de existentes y describan el proceso actual y la opción de cambiarlo.

Como segunda fuente se verificarán las bases de datos científicas especializadas en ingeniería para aplicar las metodologías necesarias en cuanto a la producción de software de calidad y que den valor agregado al producto final enmarcadas en procesos conocidos y de trazabilidad comprobable. Además de las generalidades legales y sociales que pueda acarrear el uso de material multimedia con propiedad intelectual.

1.9.4. Tratamiento de la información

La información recopilada a través de la aplicación de la encuesta será anónima sin causar perjuicio a los originadores de la misma. Luego de ser obtenida se analizará para seleccionar las coincidencias importantes y generar las directrices de modelamiento del producto de software que se planea construir. Por otro lado la documentación técnica guía para el desarrollo del software será claramente referenciada en la bibliografía del producto dando el valor a las mismas y respetando los principios de propiedad intelectual.

1.10. Organización del trabajo de grado

Parte II

DESARROLLO DE LA INVESTIGACIÓN

Capítulo 2

Recolección y ordenamiento de la información

2.1. Encuesta

2.2. Tabulación y ordenamiento de información

2.3. Análisis de la información

Capítulo 3

Fases del diseño del prototipo

3.1. Arquitectura Empresarial

3.1.1. Aplicación

Comportamiento de Aplicación

El punto de vista Comportamiento de la aplicación describe el comportamiento interno de una aplicación; Por ejemplo, ya que realiza uno o más servicios de aplicación. Este punto de vista es útil para diseñar el comportamiento principal de las aplicaciones o para identificar la superposición funcional entre diferentes aplicaciones.

Modelo

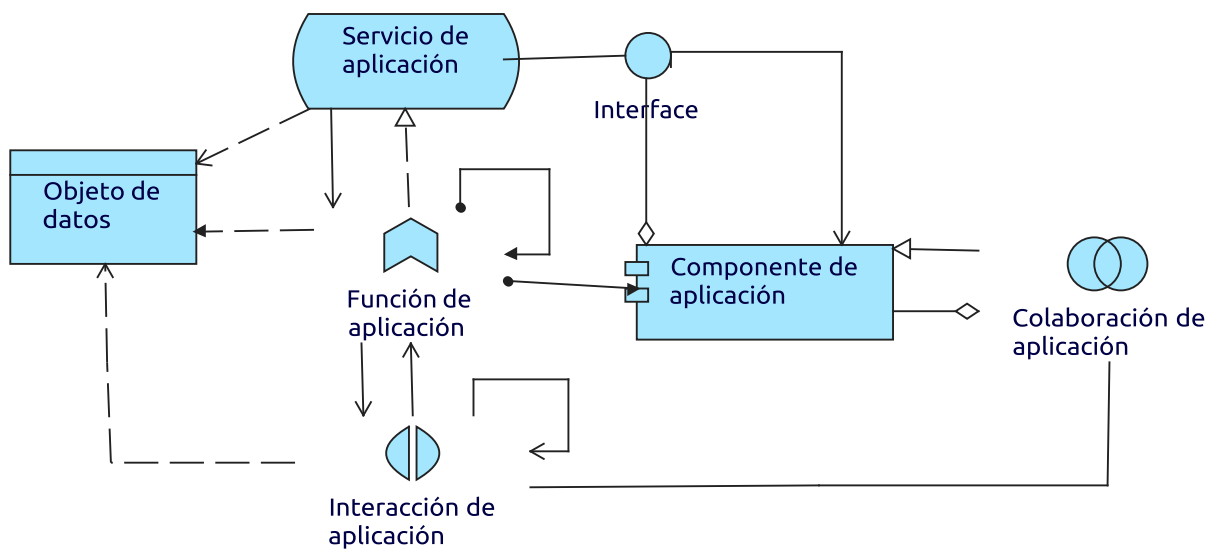
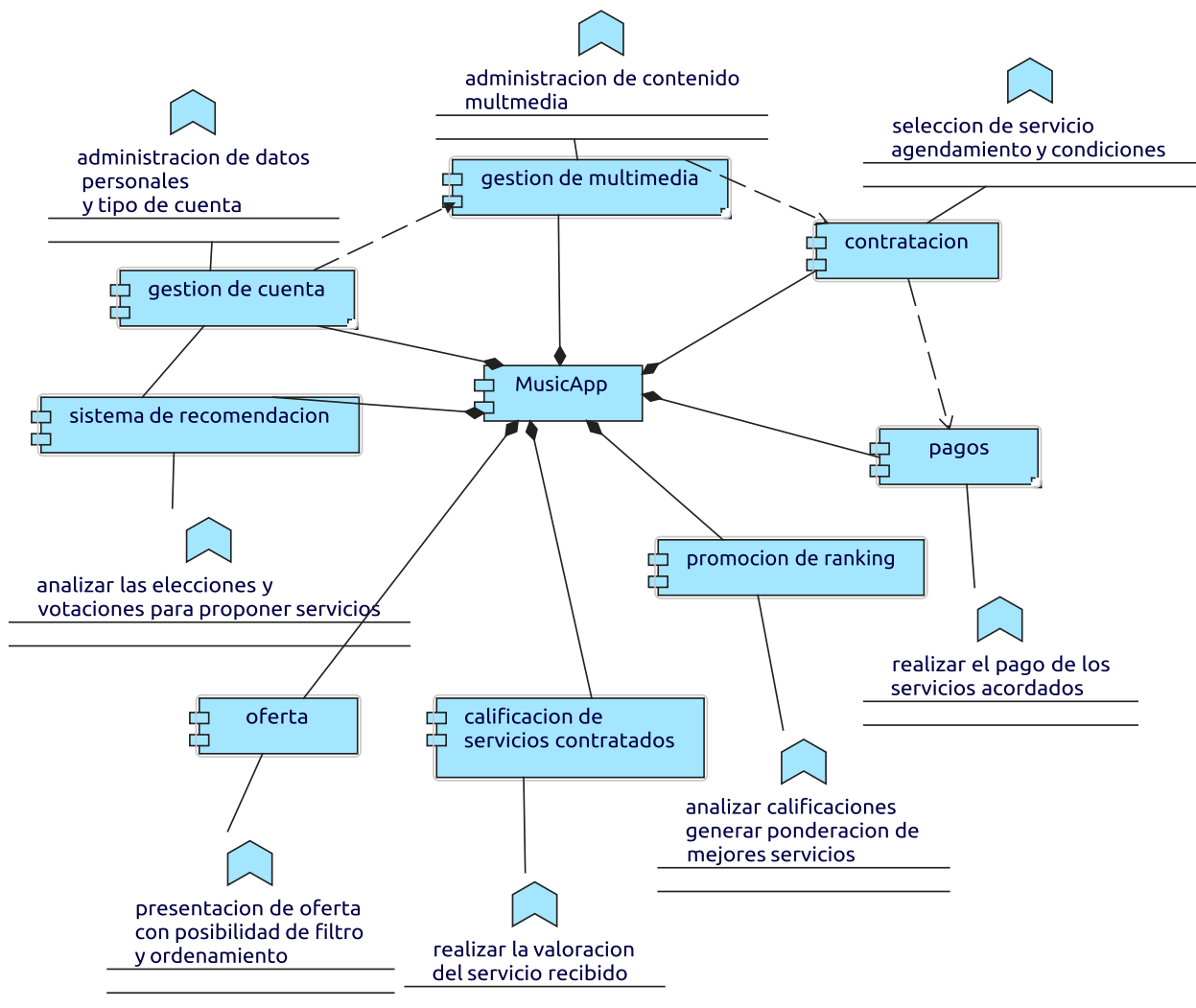


Figura 3-1: Organización

Caso de Estudio**Figura 3-2: Caso**

Cooperación de Aplicación

El punto de vista de la Cooperación en la Aplicación describe las relaciones entre los componentes de las aplicaciones en términos de los flujos de información entre ellos, o en términos de los servicios que ofrecen y utilizan. Este punto de vista se suele utilizar para crear una visión general del panorama de aplicaciones de una organización. Este punto de vista también se utiliza para expresar la cooperación (interna) u orquestación de servicios que en conjunto apoyan la ejecución de un proceso de negocios. Modelo

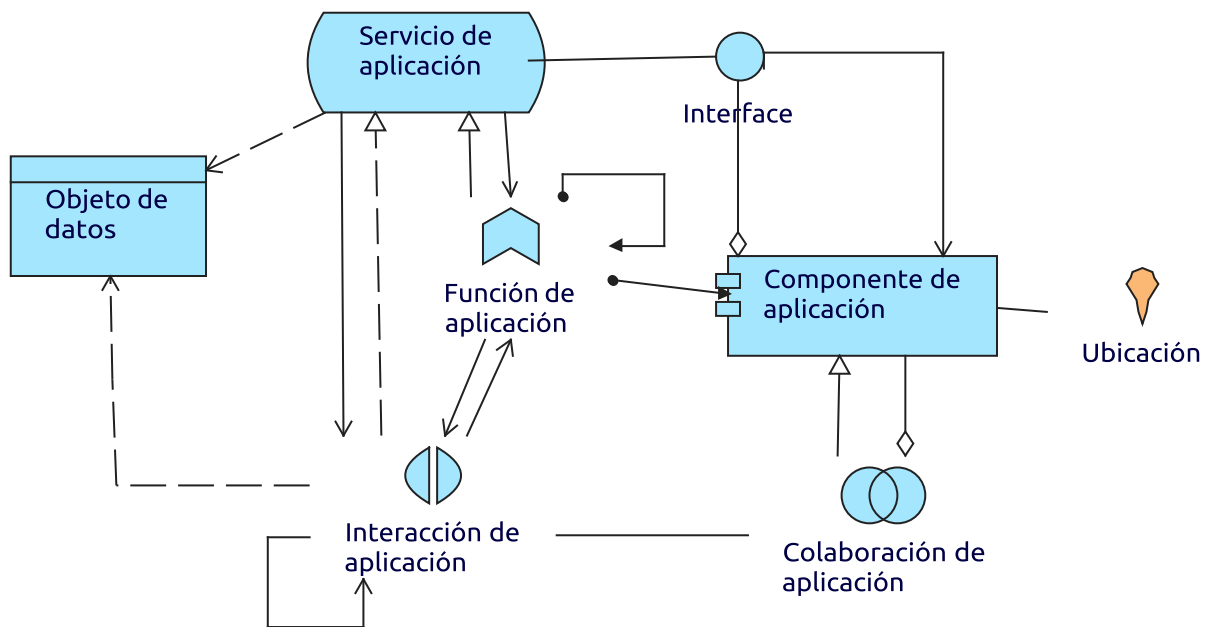
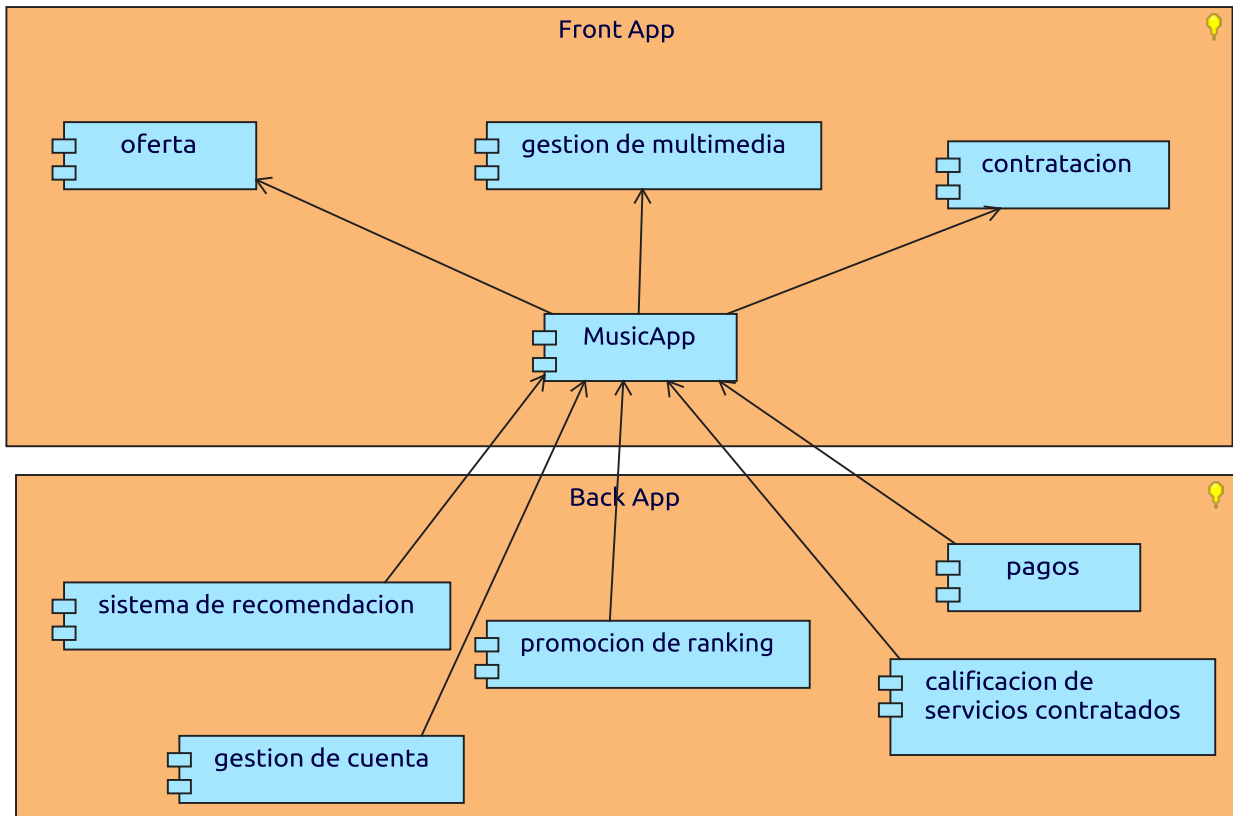


Figura 3-3: Organización

Caso de Estudio

**Figura 3-4:** Caso

Estructura de Aplicación

El punto de vista de la Estructura de la aplicación muestra la estructura de una o más aplicaciones o componentes. Este punto de vista es útil para diseñar o comprender la estructura principal de las aplicaciones o componentes y los datos asociados; por ejemplo, para desglosar la estructura del sistema en construcción o para identificar componentes de aplicaciones heredadas que sean adecuados para la migración / integración. Modelo

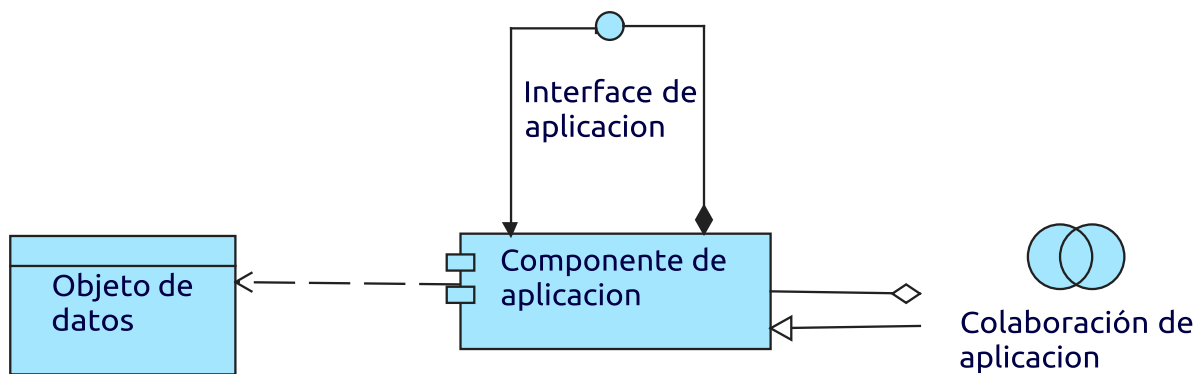


Figura 3-5: Organización

Caso de Estudio

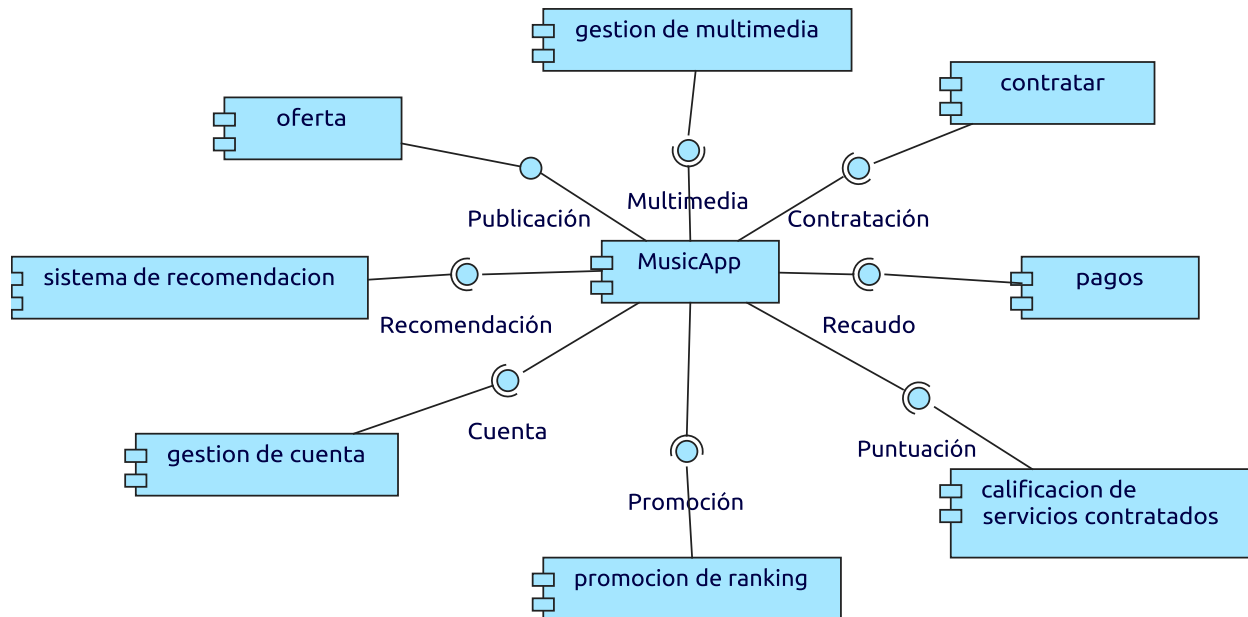


Figura 3-6: Caso

Código

```

1000 package inv.logica;
1002 public interface IContratacion {
1004     void seleccionarServicio();
1004     void agendar();
1004 }
  
```

Desarrollo/ArquitecturaEmpresarial/Aplicacion/fuentes/IContratacion.java

```

1000 package inv.logica;
1002 public interface ICuenta {
1004     void datosPersonales();
1004     void tipoDeCuenta();
1004 }
  
```

Desarrollo/ArquitecturaEmpresarial/Aplicacion/fuentes/ICuenta.java

```

1000 package inv.logica;
1002 public interface IMultimedia {
1004     void contenidoMultimedia();
1004 }
  
```

Desarrollo/ArquitecturaEmpresarial/Aplicacion/fuentes/IMultimedia.java

```
1000 package inv.logica;  
1002 public interface IPromocion {  
    void analizarCalificaciones();  
1004    void generarPonderacionServicios();  
    }
```

Desarrollo/ArquitecturaEmpresarial/Aplicacion/fuentes/IPromocion.java

```
1000 package inv.logica;  
1002 public interface IPublicacion {  
    void publicarOferta();  
1004    void obtenerOfertas();  
    }
```

Desarrollo/ArquitecturaEmpresarial/Aplicacion/fuentes/IPublicacion.java

```
1000 package inv.logica;  
1002 public interface IPuntuacion {  
    void realizarValoracionServicio();  
1004 }
```

Desarrollo/ArquitecturaEmpresarial/Aplicacion/fuentes/IPuntuacion.java

```
1000 package inv.logica;  
1002 public interface IRecaudo {  
    void realizarPago();  
1004 }
```

Desarrollo/ArquitecturaEmpresarial/Aplicacion/fuentes/IRecaudo.java

```
1000 package inv.logica;  
1002 public interface IRecomendacion {  
    void analizarElecciones();  
1004    void votacionesProponerServicios();  
    }
```

Desarrollo/ArquitecturaEmpresarial/Aplicacion/fuentes/IRecomendacion.java

Uso de Aplicación

El punto de vista de Uso de la aplicación describe cómo se usan las aplicaciones para admitir uno o más procesos de negocios, y cómo las usan otras aplicaciones. Se puede usar para diseñar una aplicación identificando los servicios que necesitan los procesos de negocios y otras aplicaciones, o para diseñar procesos de negocios describiendo los servicios disponibles. Además, dado que identifica las dependencias de los procesos de negocios con respecto a las aplicaciones, puede ser útil para los gerentes operativos responsables de estos procesos. Modelo

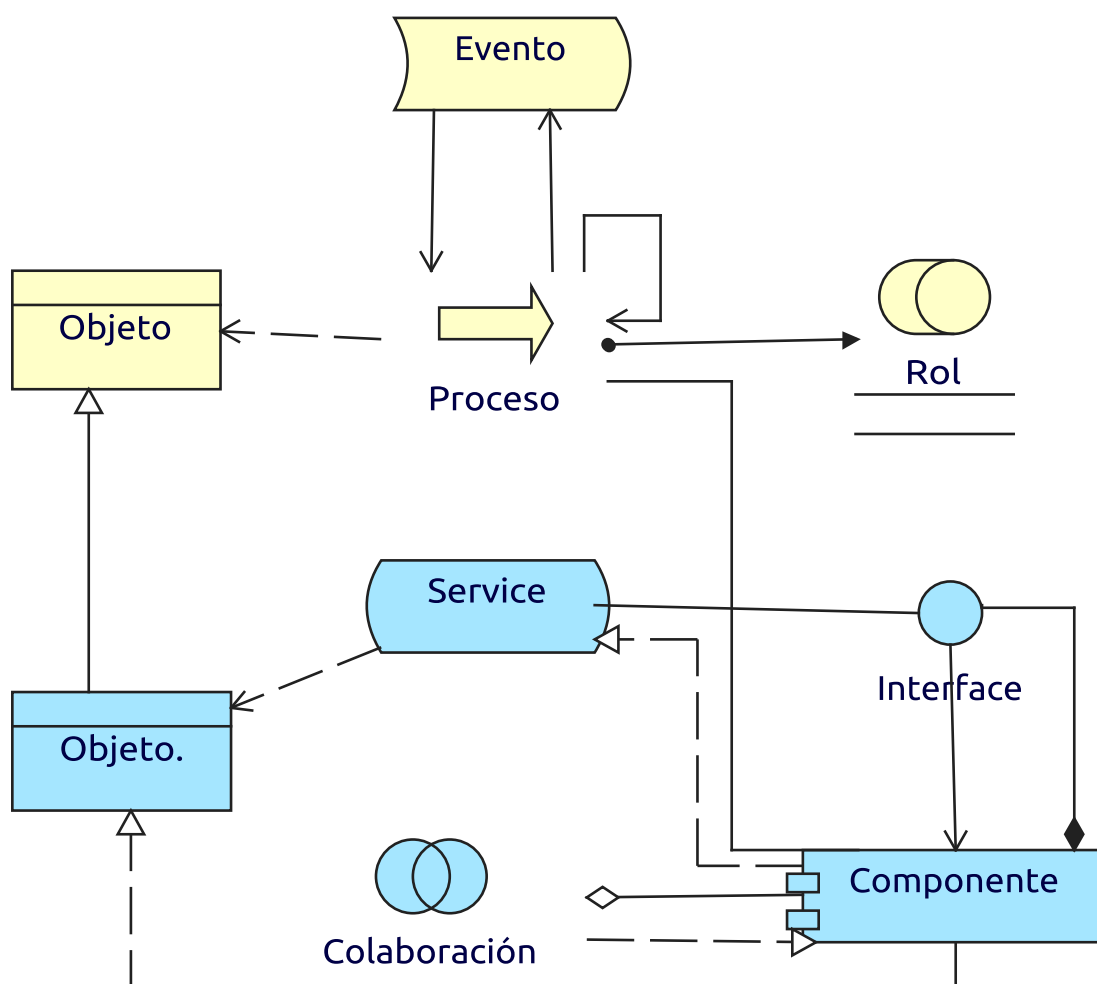


Figura 3-7: Organización

Caso de Estudio

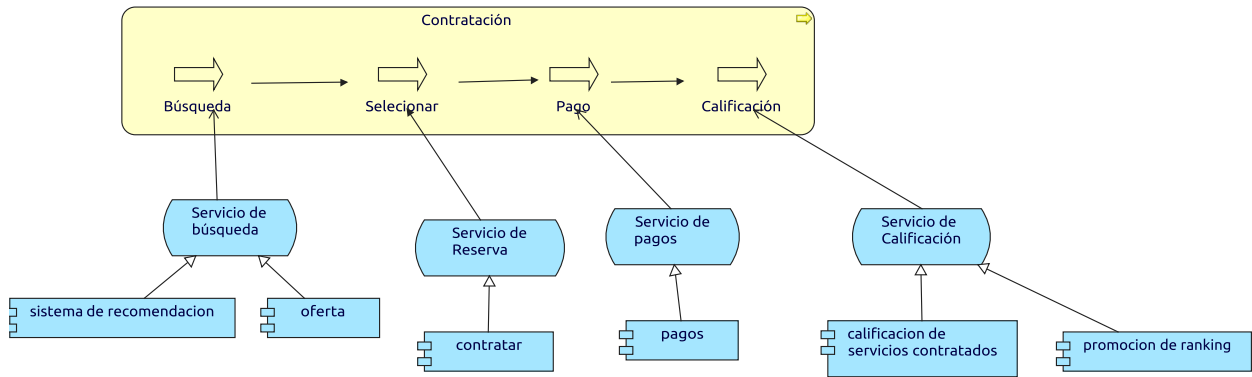


Figura 3-8: Caso

3.1.2. Tecnología

Infraestructura

El punto de vista de Infraestructura contiene los elementos de infraestructura de software y hardware que soportan la capa de aplicación, como dispositivos físicos, redes o software del sistema (por ejemplo, sistemas operativos, bases de datos y middleware). Modelo

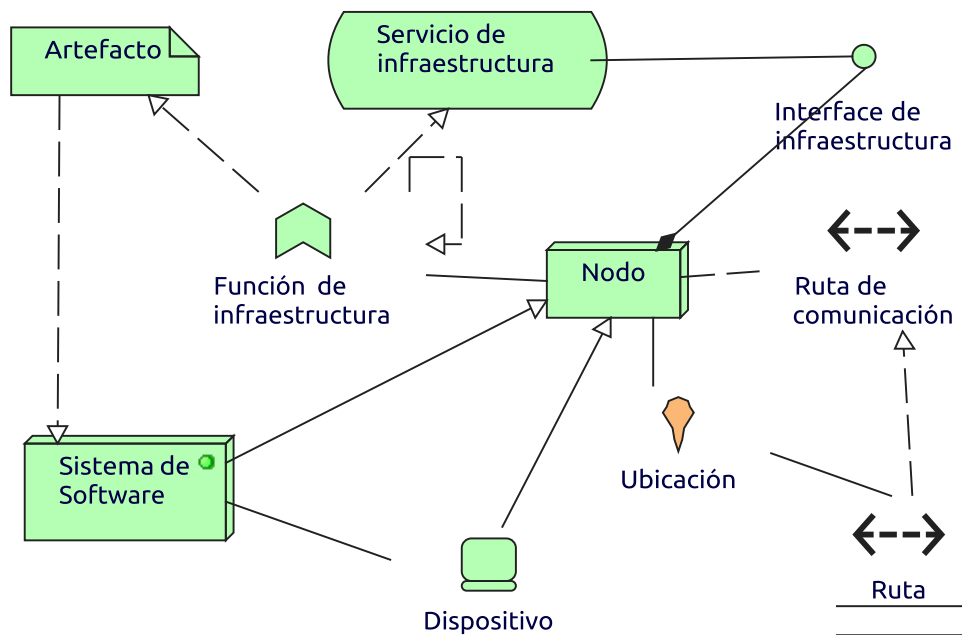
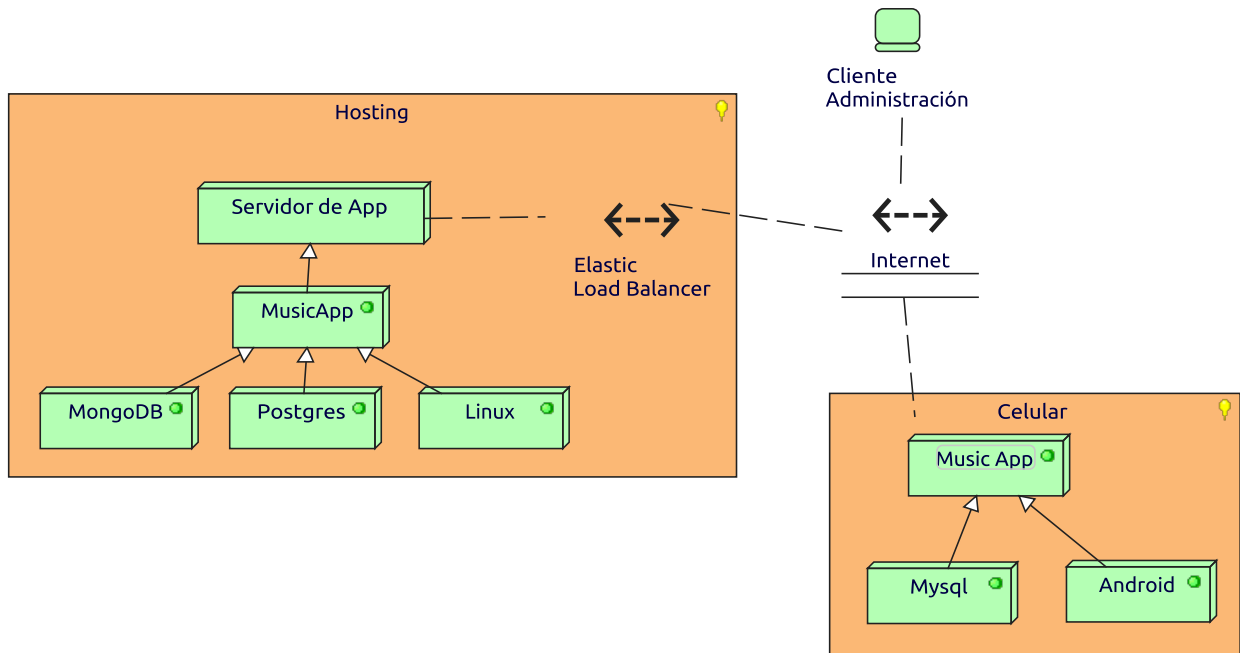
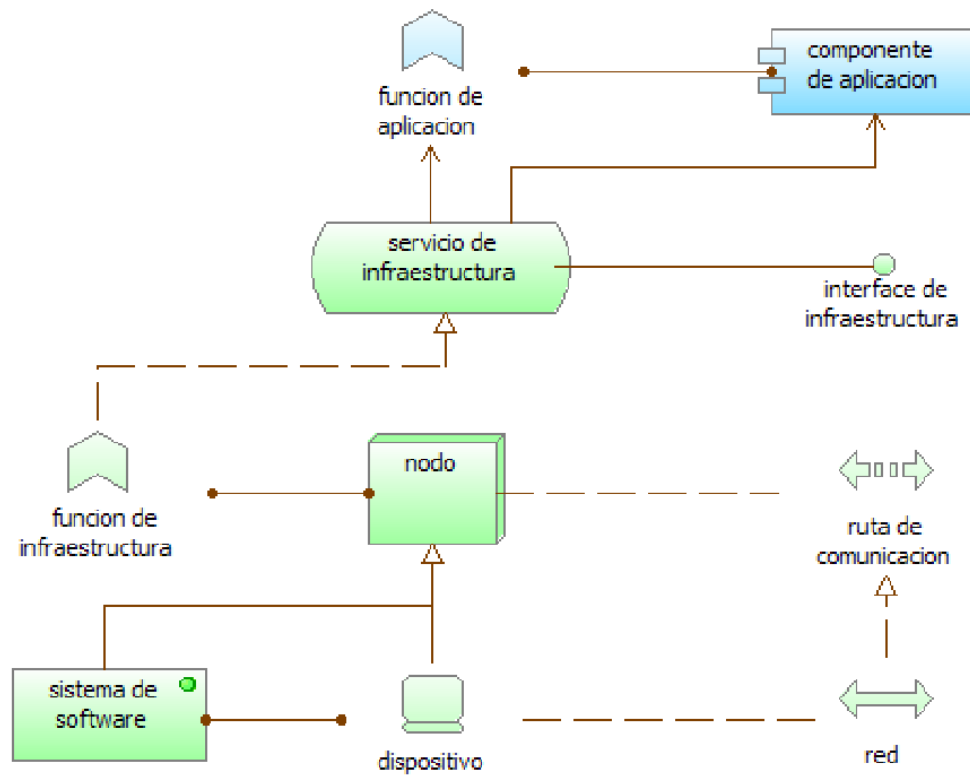


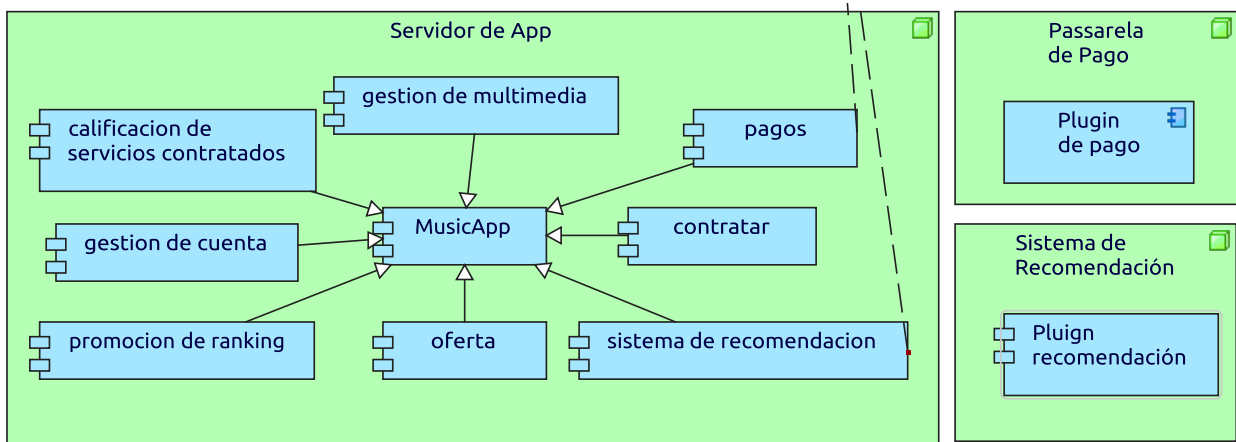
Figura 3-9: Organización

Caso de Estudio

**Figura 3-10:** Caso

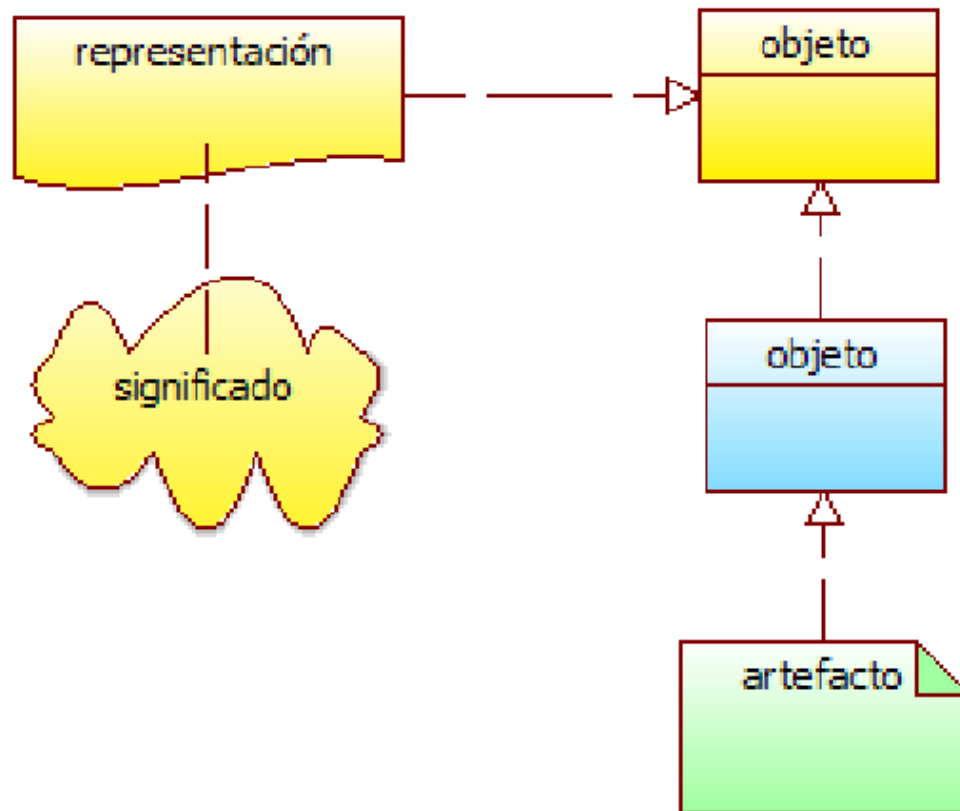
Uso de Infraestructura**Modelo****Figura 3-11:** Organización

Caso de Estudio

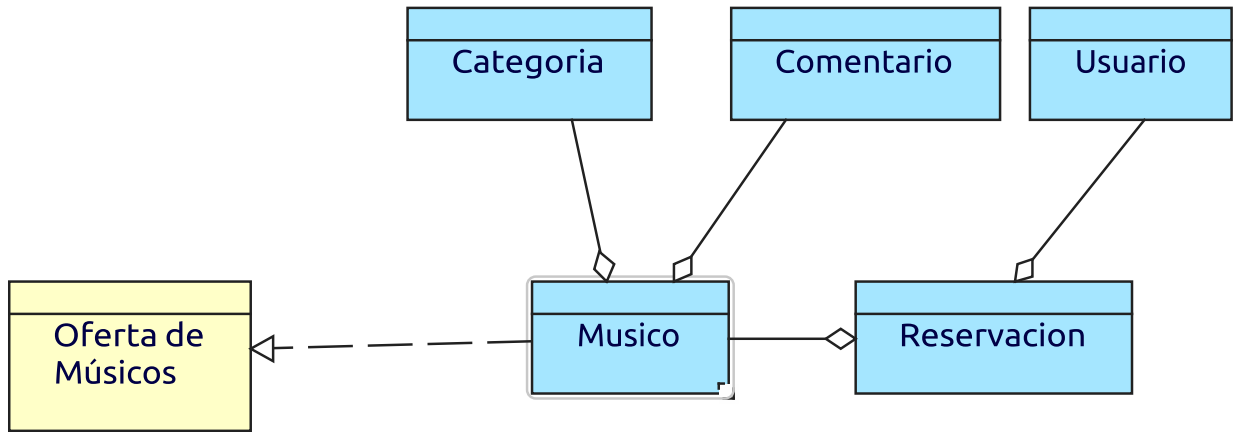
**Figura 3-12:** Caso

Estructura de Información

Modelo

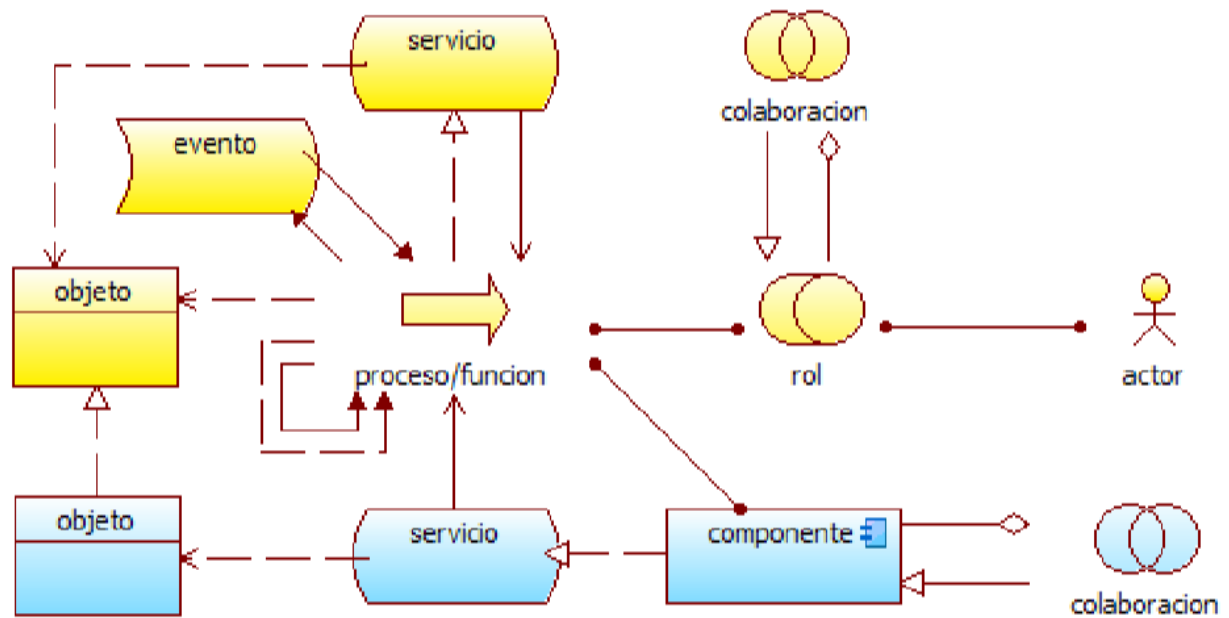
**Figura 3-13:** Organización

Caso de Estudio

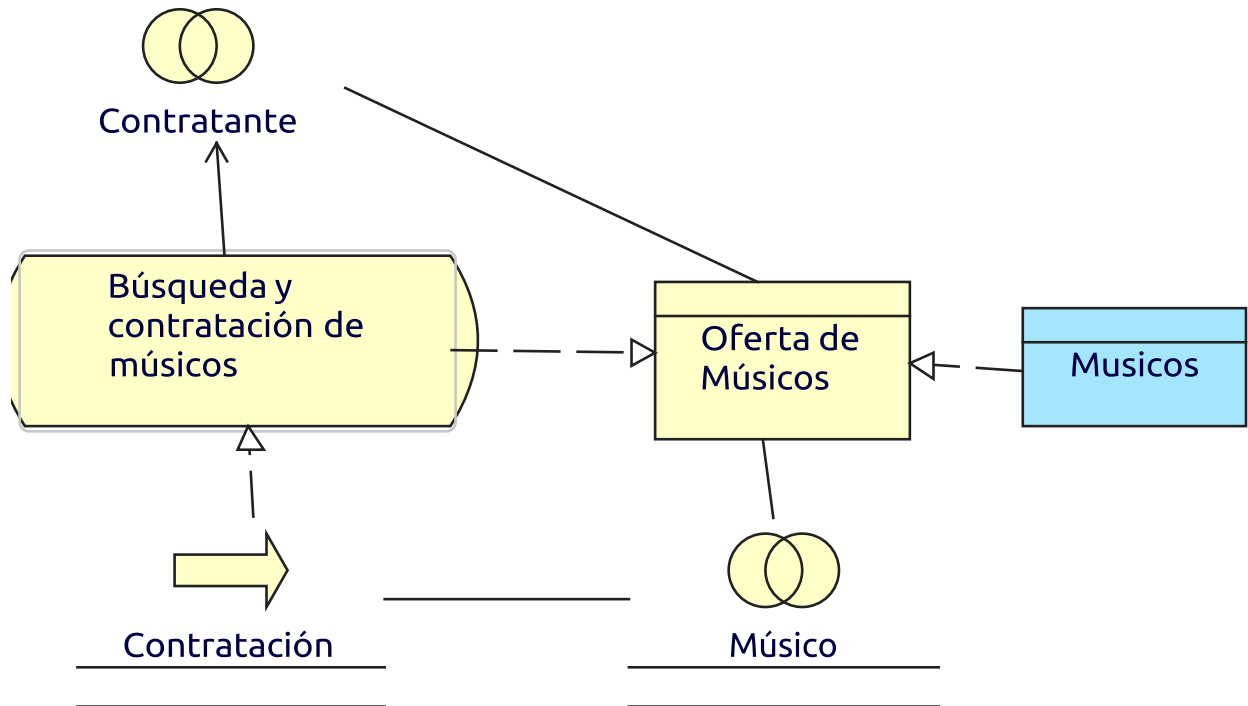
**Figura 3-14:** Caso

Realización del servicio

Modelo

**Figura 3-15:** Organización

Caso de Estudio

**Figura 3-16:** Caso

Capas

Modelo Todos los conceptos y todas las relaciones. Caso de Estudio

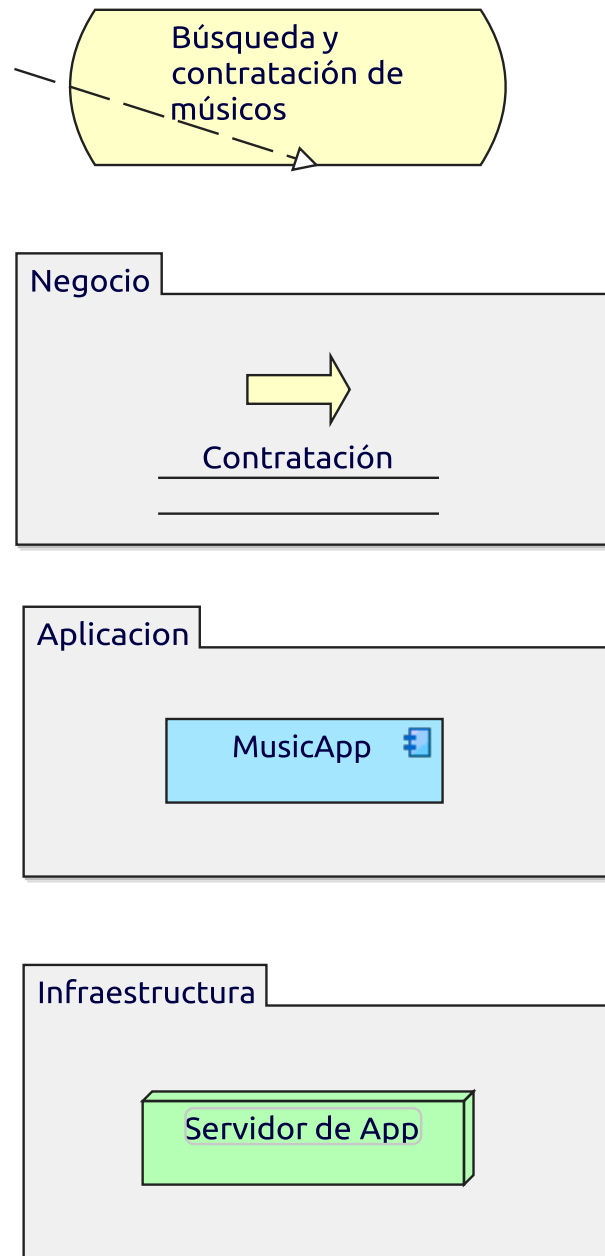


Figura 3-17: Caso

Capítulo 4

Motivación

4.1. Historias de usuario

4.1.1. Sesión

SYS-APP-001: Registro

Módulo	Submódulo:
Sistema	Sesión
Texto Descriptivo	
Yo como usuario sin registrar deseo registrarme en el sistema para empezar a disfrutar de las opciones del sistema	
Datos de Entrada	Datos de Salida
<ul style="list-style-type: none">■ Nombre■ Apellido■ Email■ Contraseña■ Tipo de cuenta: músico o contratante 0em	
Criterios de aceptación	
<ul style="list-style-type: none">■ Escenario 1: El usuario está registrado.<ul style="list-style-type: none">● mostrar un mensaje que indique que el usuario ya está registrado.■ Escenario 2: El usuario no está registrado.<ul style="list-style-type: none">● mostrar un mensaje de registro exitoso.	

SYS-APP-002: Login

Módulo	Submódulo:
Sistema	Sesión
Texto Descriptivo	
Yo como usuario registrado deseo hacer login en el sistema para ingresar a las opciones del sistema	
Datos de Entrada	Datos de Salida
<ul style="list-style-type: none"> ▪ Email ▪ Contraseña 	
Criterios de aceptación	
<ul style="list-style-type: none"> ▪ Escenario 1: Usuario no registrado. <ul style="list-style-type: none"> • Mostrar mensaje de usuario no registrado • Invitar al usuario a registrarse ▪ Escenario 2: La contraseña o el mail no coincide. <ul style="list-style-type: none"> • Mostrar mensaje de error de usuario y/o contraseña. ▪ Escenario 3: Login exitoso. <ul style="list-style-type: none"> • Direccionar al usuario a GOTO. ▪ Escenario 4: Error de red. <ul style="list-style-type: none"> • Mostrar mensaje de error de red, que revise las conexiones a internet. 	

SYS-APP-003: Cerrar Sesión

Módulo	Submódulo:
Sistema	Sesión
Texto Descriptivo	
Yo como usuario con login deseo cerrar sesión para no tener acceso a mis datos	
Datos de Entrada	Datos de Salida
Criterios de aceptación	

SYS-APP-004: Mi perfil

Módulo	Submódulo:
Sistema	USuario
Texto Descriptivo	
Yo como usuario con login deseo ver la información de mi perfil	
Datos de Entrada	Datos de Salida
<ul style="list-style-type: none"> Id Usuario 	<ul style="list-style-type: none"> Nombre Apellido Correo Foto
Criterios de aceptación	

SYS-APP-005: Cambiar Contraseña con login

Módulo	Submódulo:
Sistema	Usuario
Texto Descriptivo	
Yo como usuario con login deseo cambiar mi contraseña	
Datos de Entrada	Datos de Salida
<ul style="list-style-type: none"> Contraseña actual Contraseña nueva 	
Criterios de aceptación	
<ul style="list-style-type: none"> Escenario 1: La contraseña actual no coincide. <ul style="list-style-type: none"> Mostrar mensaje de contraseña inválida Escenario 2: La contraseña actual es igual a la nueva. <ul style="list-style-type: none"> Mostrar mensaje de error, la contraseña nueva no puede ser igual a la actual. Escenario 3: Cambio exitoso. <ul style="list-style-type: none"> Mostrar mensaje de cambio exitoso Direccionar al usuario a la vista principal. 	

4.1.2. Contratante

SYS-APP-CNT-001: Contratante: Listar músicos

Módulo	Submódulo:
Sistema	Usuario
Texto Descriptivo	
Yo como contratante con login deseo ver los músicos disponibles y activos en la app para poder contactarlo	
Datos de Entrada	Datos de Salida
<ul style="list-style-type: none"> ■ Filtros <ul style="list-style-type: none"> ● Categoría ● Nombre del grupo ● Ubicación 	<ul style="list-style-type: none"> ■ Nombre del grupo ■ Imagen ■ Calificación ■ Categoría
Criterios de aceptación	
<ul style="list-style-type: none"> ■ Los filtros son opcionales, se puede digitar 1 o todos ■ Se debe listar paginados 	

SYS-APP-CNT-002: Contratante: detalle músico

Módulo	Submódulo:
Sistema	Usuario
Texto Descriptivo	
Yo como contratante con login deseo ver el detalle de un músico para contactar	
Datos de Entrada	Datos de Salida
<ul style="list-style-type: none"> ■ id músico 	<ul style="list-style-type: none"> ■ Nombre del grupo ■ Descripción ■ Imagen ■ Calificación ■ Categoría ■ Comentarios ■ Precio
Criterios de aceptación	

SYS-APP-CNT-003: Contratante: Ver comentarios

Módulo	Submódulo:
Sistema	Usuario
Texto Descriptivo	
Yo como contratante con login deseo ver los comentarios realizados a un músico en específico	
Datos de Entrada	Datos de Salida
	<ul style="list-style-type: none"> ■ Foto Usuario que comenta ■ Nombre de usuario que comenta ■ Comentario ■ Fecha
Criterios de aceptación	
<ul style="list-style-type: none"> ■ Deben ir organizados del más reciente al ■ Se debe listar paginados 	

SYS-APP-CNT-004: Contratante: Reservar músico

Módulo	Submódulo:
Sistema	Usuario
Texto Descriptivo	
Yo como contratante con login deseo reservar los servicios de un músico	
Datos de Entrada	Datos de Salida
<ul style="list-style-type: none"> ■ Fecha Inicio ■ Fecha Fin 	
Criterios de aceptación	
<ul style="list-style-type: none"> ■ Escenario 1: El artista no está disponible para las fechas indicadas. <ul style="list-style-type: none"> ● Mostrar mensaje de error, y solicitar que ingrese una nueva fecha ■ Escenario 2: Si esta disponible el artista. <ul style="list-style-type: none"> ● Mostar mensaje de reserva exitosa. 	

4.1.3. Música

SYS-APP-MSC-001: Música: Ver perfil

Módulo	Submódulo:
Sistema	Música
Texto Descriptivo	
Yo como músico deseo ver mi información de perfil	
Datos de Entrada	Datos de Salida
<ul style="list-style-type: none">■ Id Músico	<ul style="list-style-type: none">■ Imágenes■ Nombre■ Descripción
Criterios de aceptación	

SYS-APP-MSC-002: Música: Editar perfil

Módulo	Submódulo:
Sistema	Música
Texto Descriptivo	
Yo como músico deseo editar mi información de perfil	
Datos de Entrada	Datos de Salida
<ul style="list-style-type: none">■ Nombre *■ Descripción *■ Valor presentación■ Visible valor	
Criterios de aceptación	
<ul style="list-style-type: none">■ Los campos con * no pueden estar vacíos	

SYS-APP-MS-003: Músico: editar imágenes

Módulo	Submódulo:
Sistema	Músico
Texto Descriptivo	
Yo como músico deseo editar las imágenes que se van a presentar al usuario	
Datos de Entrada	Datos de Salida
<ul style="list-style-type: none">Id Músico	<ul style="list-style-type: none">Imágenes
Criterios de aceptación	
<ul style="list-style-type: none">Puede tener máximo 8 imágenesPuede adicionar y eliminar imágenes	

SYS-APP-MS-004: Músico: editar video

Módulo	Submódulo:
Sistema	Músico
Texto Descriptivo	
Yo como músico deseo editar el video que se va a presentar al usuario	
Datos de Entrada	Datos de Salida
<ul style="list-style-type: none">Id Músico	<ul style="list-style-type: none">Url video
Criterios de aceptación	
<ul style="list-style-type: none">Solo puede tener un video activo	

SYS-APP-MS-005: Músico: Ver comentarios

Módulo	Submódulo:
Sistema	Músico
Texto Descriptivo	
Yo como músico deseo ver los comentarios realizados a mi grupo	
Datos de Entrada	Datos de Salida
	<ul style="list-style-type: none">■ Foto Usuario que comenta■ Nombre de usuario que comenta■ Comentario■ Fecha
Criterios de aceptación	
<ul style="list-style-type: none">■ Deben ir organizados del más reciente al más antiguo■ Se debe listar paginados	

4.2. Modelo de datos

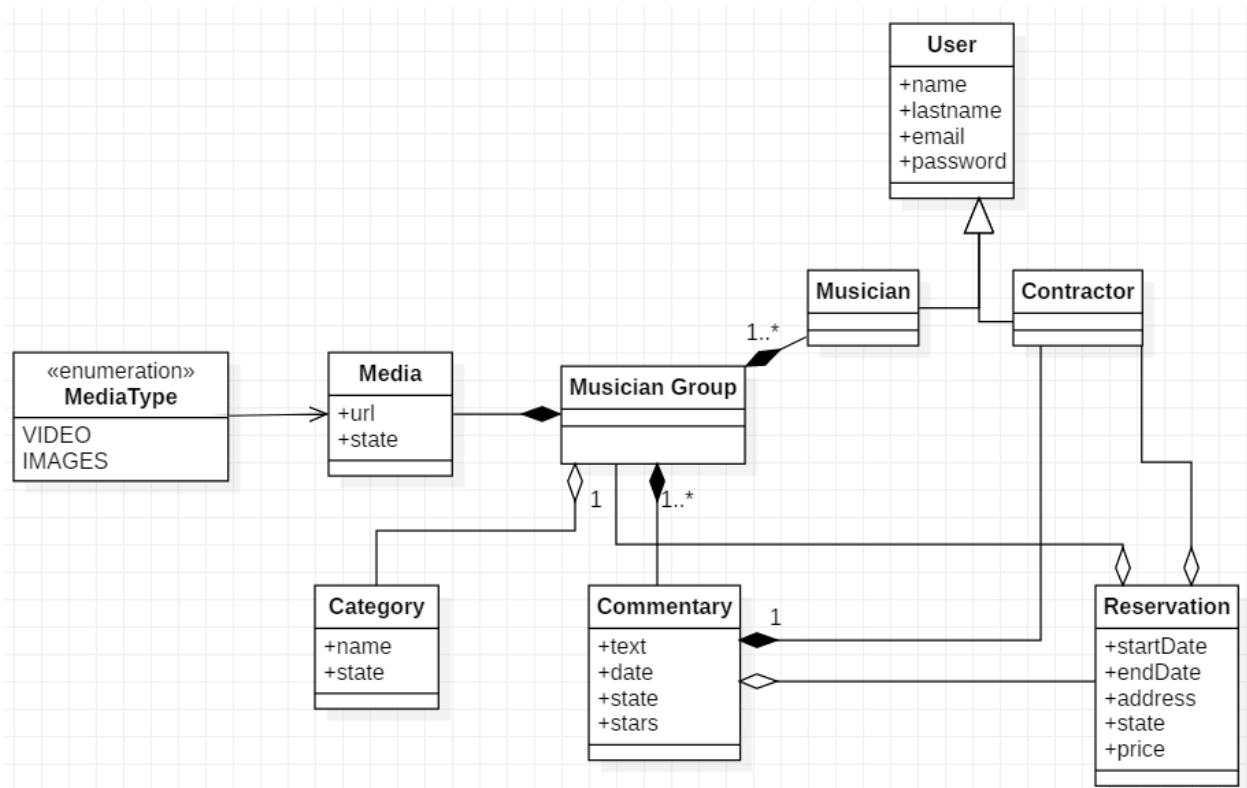


Figura 4-1: Modelo de datos.

4.3. Interfaces de usuario

Capítulo 5

Desarrollo de funcionalidades del prototipo

Parte III

CIERRE DEL PROYECTO

Capítulo 6

Resultados

Capítulo 7

Conclusiones

Capítulo 8

Prospectiva del trabajo de grado

Capítulo 9

Líneas de investigación futuras

Capítulo 10

Trabajos de Investigación futuros