Parcial 2:

Construcción de Aplicaciones Móviles; Análisis Retro Music Player

* Introducción:

La aplicación Retro Music Player, es una aplicación Open Source para Android, disponible en el Play Store. Su función principal consiste en servir como un reproductor de archivos de audio, acompañado por funcionalidades variadas de organización, categorización y parametrización de los archivos musicales. Finalmente cuenta también con segmentos de analítica individual, los cuales pueden ser accedidos por el usuario, a fin de extraer patrones de uso de sus propios archivos.

Esta aplicación, creada por Hermanth Savarala localizado en Bangalore, India; cuenta con más de un millón de descargas, así como una calificación de 4,6. Se puede afirmar pues, que es una aplicación que ha tenido un existo sustancial, a la vez de respuestas generalmente positivas de sus usuarios. Además de esto, es una aplicación bastante activa, dado que su última versión fue publicada el 12 de enero del 2020, mientras que la última publicación en su repositorio fue el 4 de mayo del mismo año (a fecha de este análisis). No queda de sobra aclarar, además que es la primera aplicación de Hermanth Savarala.

A través del presente escrito, nos gustaría hacer un análisis a profundidad de cada uno de los aspectos de desarrollo de esta aplicación. Desde su diseño, hasta su implementación, llegando incluso hasta sus presentes actualizaciones. Para llevar a cabo dicho análisis, es menester, en primer lugar, dar claridad a las distintas funcionalidades con las que cuenta esta aplicación, a fin de poder entender a fondo su desarrollo.

Retro Music Player, es un reproductor de música enfocado hacia la simplificación, la confiabilidad y el diseño intuitivo de sus funcionalidades. Como bien su nombre indica, este reproductor se especializa en evitar las complicaciones para el usuario. A este respecto sus funcionalidades hacen eco de este enfoque, facilitándole al usuario servicios clave, de tal forma que el usuario nunca pierda la sensación de familiaridad y simpleza. A juicio tal, esta aplicación cumple con todas las funcionalidades esperadas de un reproductor de música moderno, como la creación de listas de reproducción, la presencia de un ecualizador, analítica personalizada y facilidades para la categorización de archivos de audio.

Ahora bien, a la hora de comenzar el análisis de esta aplicación, hemos de empezar por la etapa de diseño. Teniendo en cuenta, los enfoques antes descritos, a la hora de entender las motivaciones detrás de las decisiones de diseño aquí plasmadas.

* Design Thinking:

Analizaremos el diseño de esta aplicación a través del paradigma del “Design Thinking” sin embargo, dada nuestra posición como terceros en el desarrollo de esta aplicación, nos enfocaremos en los primeros tres puntos de dicho paradigma (Empatizar, Definir e Idear).

1. Empatizar:

A la hora de deducir el proceso de empatización que llevo al desarrollo de esta aplicación, consideramos importante regresar al enfoque de la aplicación, es decir la simplificación de sus funcionalidades (desde el punto de vista del usuario).

Desde esta perspectiva el problema a resolver se hace evidente, la frustración y la sobre-saturación de funcionalidades en los reproductores de música modernos. Los cuales incluyen un sinfín de opciones de categorización y, más importante aún, de ecualización.

Estas funcionalidades, pueden en gran medida, aportar mucho valor a una aplicación. Sin embargo, para aquellos sin los recursos intelectuales para aprovechar estas funcionalidades pueden rápidamente tornarse en un estrobo. Especialmente nos hemos de referir a la función de ecualización, la cual, en aplicaciones modernas, tiende a incluir un gran número de parámetros relacionados con el output de sonido. Parámetros que no todo usuario entiende (ni quiere entender).

1. Definir:

Es claro, entonces, el problema a resolver. Existe una sobre-oferta de funcionalidades, que en tiempos presentes pueden llegar a considerarse un estorbo, afectando negativamente el valor percibido en una aplicación, desde el punto de vista de ciertos usuarios.

Esta sobre-oferta abre la posibilidad de, sin perder la calidad esperada de un reproductor, ni sus capacidades de personalización, ofrecer una alternativa que refleje la familiaridad anterior de un reproductor de música. Esto en aras de evitar la sobre-complejidad presente en muchas aplicaciones modernas.

1. Idear:

Consideramos que, a raíz de los planteamientos anteriormente mencionados, surgió la idea de crear Retro Music Player. La cual aporta las mismas funcionalidades que se esperan de un reproductor musical moderno, más mantiene al usuario alejado de parametrizaciones complejas, intentando asemejarse lo más posible a un reproductor de música original.

Para alcanzar esta meta, se debía diseñar una aplicación, capaz de manejar la parametrización moderna de manera interna, sacrificando así la personalización extrema, presente en las aplicaciones modernas. Sin mencionar la complejización añadida al proceso de desarrollo y programación. Sin embargo, accediendo a un sector del mercado interesado en este tipo de aplicaciones, simples.

No será suficiente además con mantener los procesos de parametrización, encapsulados de forma interna. También se deben enfatizar en el diseño de los componentes de la aplicación, decisiones de diseño que promuevan la sencillez y la navegación intuitiva de la aplicación.

1. No sobra decir, que la app no contiene facilidades para individuos que sufran de invalidez, como ceguera.

* Tipo de Implementación:

La app Retro Music Player, es una app Nativa de Android, escrita en Kotlin y Java al unísono. No tiene, sin embargo, funcionalidades sustanciales nativas que hagan referencia a los sensores del dispositivo, incluyendo ubicación y cámara, entre otras. Únicamente toma recursos del almacenamiento interno del celular, y accede a su output de sonido.

1. Java & Kotlin:

Si bien estos lenguajes son muy similares, es una buena práctica a nivel de programación el evitar programar en distintos lenguajes sin encapsulamiento, más aún si estos lenguajes no cumplen funciones distintas y forman parte de un mismo componente programático. Si bien es factible que su naturaleza como una aplicación Open Source, permita a los colaboradores añadir código de una forma más bien poco restringida, consideramos que, en aras de un escalamiento saludable, no sobraría hacer un proceso de conversión entre ambos lenguajes. Este proceso podría incrementar la solidez y confiabilidad del desarrollo. No está de más aclarar que la conversión automática entre ambos lenguajes existe y bien debería ser tomada en cuenta a futuro, así como regulaciones un tanto más estrictas sobre las colaboraciones al repositorio.

1. ¿Nativa/Cross-Platform?

Retro Music Player, es una aplicación nativa para Android. Esto quiere decir que no cuenta con ninguna contraparte en IOS, ni una versión web; desarrollo con el cual estamos en desacuerdo, dada la naturaleza de la aplicación.

Como ya hemos expuesto, Retro Music Player intenta abarcar un segmento de mercado especifico, segmento cuya identidad es completamente independiente a la plataforma de desarrollo. Más allá de esto, por su propia naturaleza esta aplicación, no requiere un uso sustancial de recursos del hardware del dispositivo; por lo que un desarrollo nativo no beneficia su desempeño de forma sustancial. Es bien entendido que el almacenamiento local es accesible desde otros métodos de desarrollo, así como los demás requerimientos de las funcionalidades de la aplicación.

Desde esta perspectiva, este desarrollo ha excluido a todo el segmento de mercado de IOS, sin ningún beneficio en retrospectiva. Esta decisión de diseño sin duda ha obstaculizado la escalabilidad y el modelo de negocio, de esta aplicación particular.

1. Deberíamos tomar en cuenta, sin embargo, las habilidades preexistentes de los desarrolladores, las cuales se enfocan exclusivamente hacia desarrollo nativo en Android. Desde esta perspectiva, un desarrollo cross-platform o incluso hibrido web, hubiera incluido una curva de aprendizaje, la cual podemos asumir, comenzaría desde lo más bajo.

Desde esta perspectiva, la relación costo-beneficio de un desarrollo nativo en Android es menos negativa, más consideramos que, dado el éxito que esta aplicación ha tenido en el Play Store, el valor potencial de un desarrollo cross-platform continúa siendo superior a su costo.

* Contexto:

Como ya hemos establecido, Retro Music Player, se enfoca en un sector de mercado especifico, con unos requerimientos funcionales particulares. Este sector de mercado, a su vez está conformado por individuos, los cuales podríamos parametrizar de acuerdo con las funcionalidades por las cuales muestran predilección, como haremos a continuación.

1. Usuario:

Podríamos simplificar sustancialmente la identidad de un usuario de esta aplicación, catalogándolos como individuos sin habilidad tecnológica. De esta forma podríamos incluso relacionarlos con individuos de edad avanzada, o de grupos socio-económicos de menor capacidad adquisitiva. Sin embargo, las métricas establecidas para esta aplicación en el PlayStore contradicen estas categorizaciones, evidenciando su poca profundización.

Mejor considero que el usuario potencial para esta aplicación es mucho más genérico, refiriéndose a un individuo que no tiene interés alguno en parametrizar su experiencia de reproducción musical. Independientemente de sus habilidades tecnológicas, este tipo de individuo desea obtener una experiencia fácil e intuitiva, a costo de su libertad de personalizar dicha experiencia (feature que probablemente, hubiera dejando intacto, de estar presente).

Este usuario ve las funciones de parametrización, en el mejor de los casos como un estorbo. Como hemos podido observar en los comentarios de la app en el PlayStore, se han dado casos en los cuales la sobre parametrización ha dañado la experiencia de reproducción de algunos individuos, talvez de forma irreversible (al no conocer medios para volver a la parametrización de fábrica). Estos individuos, por ende, tienden a ignorar e incluso a evitar estas parametrizaciones que consideran innecesarias a gran escala.

1. Ambiente de desarrollo:
2. Features:

Las funciones de categorización (“edición de etiquetas”) existen para facilitar con el ordenamiento y la navegación a través de los archivos, sin embargo, no su utilización no es mandatoria, y los componentes que la facilitan son sencillos. Esto no permite tener un alto número de etiquetas para categorizar los archivos en múltiples categorías simultáneamente como en otras aplicaciones.

Las funciones de reproducción, permiten al usuario determinar la forma en la cual la aplicación habrá de reproducir los archivos de audio. Estas funcionalidades incluyen el orden de reproducción, y la ecualización del audio. Ambas se han mantenido a un margen de facilidad extremo. La ecualización, simplemente actúa como una conexión con las opciones de ecualización presentes en el menú de configuración del dispositivo.

Las funciones de analítica, mantienen registro de algunas métricas claves, sin necesidad de intervención por el usuario. Posteriormente estos resultados (archivos más escuchados, historial de archivos, etc.) estarán disponibles para el usuario, a fin de facilitar su utilización de los servicios de reproducción.

Existen, además, funciones de personalización del UI. Estas funciones usualmente provienen de actualizaciones, en modo de repuesta a usuarios deseando una experiencia de personalización más generosa. Sin embargo, dichos parámetros de personalización se encuentran alejados de las funcionalidades principales de la aplicación.

Finalmente, la aplicación accede al output de sonido del dispositivo. Estos recursos son utilizados de forma prioritaria por la aplicación, de tal forma que, aun si se ejecuta un cambio de actividad, relegando Retro Música Player a un segundo plano, la aplicación continuara ejecutando la reproducción en el background.

1. Dispositivo:

Esta aplicación utiliza de manera exclusiva el almacenamiento local del dispositivo Android. No cuenta con funcionalidades web, ni almacenamiento externo. Estas restricciones se aplican tanto a los archivos de audio como a los elementos propios de la aplicación como listas de reproducción, o etiquetas. Por esta razón la aplicación debe obtener el permiso del usuario para acceder a los archivos del almacenamiento local.

La aplicación tampoco consume recursos de los sensores propios del dispositivo de manera notable, y únicamente accede a los servicios del UI y el output de sonido.

* Analítica:

La analítica individual, presentada como un feature para uso del usuario, cuenta con la siguiente información.

1. Historial: Mostrando un historial de los archivos escuchados.
2. Añadidos Recientemente: Enseña un historial, no de archivos escuchados, sino archivos añadidos al almacenamiento local del dispositivo.

1. Mas Reproducidos: Genera una lista ordenada con los archivos más frecuentes.
2. Mas Reproducciones por Etiqueta: Genera una lista ordenada, por cada etiqueta, con las etiquetas más escuchadas (ej. álbumes más escuchados)

Estas funciones, sin embargo, no están directamente ligadas a funciones de analítica para los desarrolladores. Existen exclusivamente como funcionalidades internas de la aplicación, enfocadas a brindarle al usuario un mejor servicio.

La fuente de análisis para los desarrolladores, en referencia a su aplicación, está ligada a contactos directos, ya sea por medio de las funciones de analítica incluidas en el PlayStore, como los contactos directos con los responsables. Sin embargo, esta alternativa da como resultado, respuestas generalmente difusas, ambiguas e incluso erróneas sobre la precepción general de la aplicación.

Se debe tomar en cuenta el perfil de un individuo dispuesto a escribir directamente a los desarrolladores sobre una aplicación, así como su intención (usualmente antagónica, frente a la aplicación). Esta aproximación puede ser generalmente nociva, dado que las respuestas de los usuarios estarán parcializadas de forma negativa, en la mayoría de los casos.

Utilizar las funcionalidades del PlayStore, tampoco es ideal, dado que no es una plataforma especializada en este tipo de servicios. Por esta razón la retroalimentación que ofrece es extremadamente superficial, y hace referencia únicamente a la aplicación como un todo, dejando de lado otras perspectivas más granulares de la misma. Podríamos incluso mencionar, que se pierde de forma definitiva un análisis de percepción de usuario específico para el producto en cuestión, recurso que puede llegar a ser extremadamente útil a la hora de retroalimentar actualizaciones futuras del software.

1. En este punto, creemos importante mencionar que, dada su naturaleza como proyecto OpenSource, esta aplicación debe esperar actualizaciones constantes y no centralizadas. Dado su énfasis en diseño intuitivo y sencillo, la adición de funcionalidades debe ser más bien mesurada; enfocándose en la pertinencia de los features añadidos. A este respecto, cualquier forma de analítica interna, que pueda servir para evaluar la percepción de usuario frente a nuevos features podría ser una herramienta invaluable.

El gran valor aportado, se puede vislumbrar, al contrastarle con la alternativa por la cual han optado los desarrolladores, es decir agrupar nuevas funciones en los “rincones” de la aplicación. Un buen ejemplo es la funcionalidad “Pausar en Silencio” la cual permite al usuario optar por pausar la reproducción de un archivo, cuando el volumen del dispositivo este en cero. Esta funcionalidad, se encuentra tras una navegación de cuatro menús de profundidad, junto con otras funcionalidades inconexas como “esquina de dialogo” (la cual permite determinar el tamaño de los cuadros en los cuales el título del archivo aparece en pantalla, a la hora de reproducirlo). Este proceso de exploración difícilmente se podrá catalogar como intuitivo, o sencillo. Consideramos que, dada la frecuencia de actualizaciones descentralizadas, la magnitud de este problema solo incrementara con el tiempo, y es menester llevar a cabo una “limpieza” en aras de mantener la identidad de la aplicación, así como su elemento de diferenciación. Funcionalidades de analítica interna, dirigidas a responder preguntas de negocio de los desarrolladores, sería un excelente primer paso.

* Escenarios:

Como toda aplicación, Retro Music Player debe enfrentarse a toda una gama de escenarios en referencia a su ejecución en un dispositivo. A este respecto, su naturaleza OpenSource, es de gran ayuda, a la hora no solo de encontrar posibles escenarios no antes considerados, sino encontrar al mismo tiempo soluciones viables.

No hemos de olvidar, sin embargo, que esta aplicación tiene un enfoque hacia garantizar una navegación simple de sus funciones, por lo que intenta minimizar procesos complejos. Por este motivo, el abanico de situaciones posibles, que puedan afectar el funcionamiento de la aplicación se reducen sustancialmente. A continuación, listamos los escenarios que pueden afectar la aplicación, así como su correspondiente reacción.

1. Se reproduce un archivo, más la configuración del output de sonido del dispositivo se encuentra silenciada o en cero, volviendo redundante la reproducción.

* La aplicación cuenta con una opción dentro de su menú de configuraciones avanzadas, la cual permite al usuario coordinar la reproducción de los archivos con la existencia de un output funcional, pausando la reproducción en caso de que no se encuentre dicha salida.

1. Un archivo se está reproduciendo, sin embargo, una funcionalidad del dispositivo interviene de forma prioritaria la utilización del output de sonido (ej. una llamada entrante).

* La función anotada en el literal anterior incluye a su vez este tipo de interrupciones.

1. El dispositivo se recalienta, poniendo en jaque la latencia de los procesos del dispositivo.

* Nuestras pruebas indican que Retro Music Player, depende de la respuesta del sistema operativo del dispositivo para cerrarse.

1. Se pierde conexión a internet.

* En caso de tener la opción “descarga automática de portadas”, la descarga no se llevará a cabo. No se le notifica al usuario en ningún caso, y la reproducción continuara inalterada.

1. La batería del dispositivo se acerca a cero.

* La aplicación no tendrá respuesta alguna, ni informará al usuario. Tampoco guardara el estado de la reproducción.

1. El dispositivo se apaga en medio de una reproducción.

* El motor de analítica individual antes descrito registrara la última reproducción (iniciada) en el sistema, pero no almacenara información sobre el estado de la reproducción.

1. Se añaden elementos con etiquetas duplicadas, creando así inconsistencias visuales, que imposibilitan la diferenciación entre algunos archivos pro parte del usuario.

* La aplicación no toma acción, ni informa al usuario de este inconveniente.
* Diseño: ¡Material Design!

Un gran énfasis, en la presentación que Retro Music Player hace para sí misma en el PlayStore, es su énfasis en seguir los principios de “Material Design”. Esta afirmación surge de la experiencia de desarrollo del equipo responsable, dado su enfoque exclusivo hacia aplicaciones Android. Dados los lineamientos propuestos por esta metáfora de diseño, consideramos una elección acertada.

Material Design se enfoca en elementos naturales, y en semejanzas con el mundo físico en el cual habitamos. A este respecto, es difícil encontrar un paradigma que más se ajuste a la meta de Retro Music Player de tener “navegabilidad intuitiva y sencilla”, dada la familiaridad que Material Design evoca.

A continuación, haremos un análisis completo de la interfaz de Retro Music Player, componente a componente, analizando su funcionalidad, así como su diseño desde la perspectiva de Material Design, en relación con las metas establecidas por la aplicación:

* Funcionalidades:

1. Menú Principal:



-------------------- ----------------------------- Estático Barra de Búsqueda I -------------------------------------------------- Scroll Menú de Usuario - ------------------------------ Analítica I - (micro-analítica) - - - - - ----------------------------- - Analítica IIa - (macro-analítica) - (Artistas) - - - - - - - ------------------------------ Analítica IIb - (macro-analítica) - (Albumes) - - - - - - - - - - - - - - - ----------------------------- - Favoritos - - - - - - - - - - - - -------------------------------------------------- Estático Reproducción Actual (opcional) - ------------------------------ Barra de Tareas --------------------------------------------------

El menú principal de la aplicación cuenta con la información del usuario, así como la analítica individual, tanto micro (archivos individuales) como macro (archivos por etiqueta) del mismo; finalmente se representa la lista de reproducción “Favoritos” y sus contenidos:

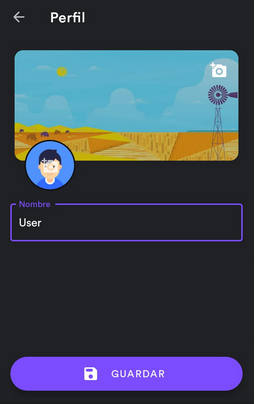
* Nota: Estos iconos son los únicos en esta pantalla, que comenzaran una reproducción inmediatamente, dado que representan archivos individuales.
* Nota: El icono “Aleatorio” mezclado entre los iconos de micro-analítica, el cual reproducirá todos los archivos de forma aleatoria. Su inclusión entre elementos de analítica podrá parecer poco intuitiva e incluso arbitraria, más consideramos que refleja de forma acertada la importancia de estas métricas a la hora de navegar entre archivos de audio con la aplicación, desde la perspectiva del usuario. Por esta misma razón estos iconos son los primeros en la jerarquía del menú principal.
* Nota: La sección de macro-analítica que hace referencia a los álbumes, es permite, al seleccionar uno de los álbumes, comenzar la reproducción del mismo. Esto se da gracias a la jerarquía de clases Artista -\*> Album

Además de esto, presenciamos la barra de búsqueda (App Bar) y la barra de tareas (Bottom Navigation Bar), elementos pseudo-omnipresentes en la aplicación.

El perfil de usuario y la analítica están relacionados a un componente con capacidad de scroll. Dichos componentes están distribuidos a modo de lista (Grid List), catalogados en relación a su función. Cada componente individual, representado por figuras curvas, emite una reacción lumínica al ser seleccionado. Adicionalmente la transición entre pantallas, sigue un camino fluido comenzando desde el extremo inferior de la pantalla, de tal forma que dicha transición se da de forma discreta, en vez de abrupta.

El micrófono en la barra de búsqueda, permite usar voz como input para la búsqueda, y es uno de los pocos ejemplos de la aplicación usando los sensores del dispositivo.

Esta vista permite, además editar el perfil ligado al dispositivo, por medio de seleccionar el icono de usuario, desplegando la siguiente vista:



Este formato permite editar el fondo, la imagen de perfil y el nombre del usuario, almacenando dichos datos de forma local en el dispositivo. La forma como los iconos se representan y se iluminan a la hora de seleccionarles siguen los lineamientos de Material Design al pie de la letra.

Al seleccionar el icono de cámara se cargará uno de los siguientes diálogos, dependiendo de a que el icono esté relacionado:

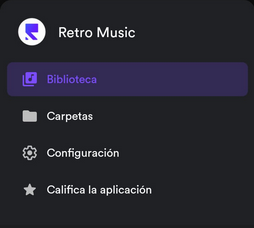


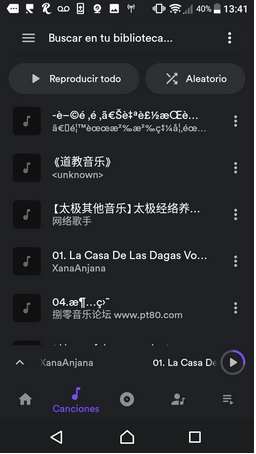
Al seleccionar la correspondiente opción “Eliminar foto de”, se volverá a la imagen de fábrica (visualizada anteriormente).

Por otro lado, al seleccionar “Nueva foto de”, se redirige al usuario al menú de selección de imágenes del dispositivo.

* Nota: las imágenes de fábrica, son detalladas y amigables, a diferencia de los iconos grises presentes en otras aplicaciones como Whatsapp. Lo cual consideramos una excelente decisión, a la hora de dar identidad a la visualización de la aplicación, aun ligada a un feature tan nimio en su desarrollo.

Al seleccionar el icono de biblioteca en la esquina superior izquierda, se desplegará el siguiente menú (Action Panels):

1. Biblioteca: La visualización principal de la aplicación.
2. Carpetas: Un explorador de archivos.
3. Configuración: Menú de parámetros de la aplicación.
4. Califica la aplicación: redirige al usuario a la función de calificación de aplicación disponible en el PlayStore.
5. Listas de Archivos I:

-------------------- ----------------------------- Estático Barra de Búsqueda II - ----------------------------- - Menú de Reproducción ------------------------------------------------- Scroll Lista - - - - - - - - - - - - - - - - - - - - -------------------------------------------------- Estático Reproducción Actual (opcional) - ------------------------------ Barra de Tareas --------------------------------------------------

La visualización de lista (List), se utiliza para representar colecciones de archivos de audio. Contiene el menú de reproducción, el cual permite comenzar la reproducción de la lista, así como el modo de reproducción (aleatorio).

Cada elemento incluye una imagen relacionada con el archivo de audio, así como su título y etiquetas pertinentes. Al seleccionar cualquier elemento, comenzara la reproducción del archivo. Finalmente, el icono de opciones despliega el siguiente menú:

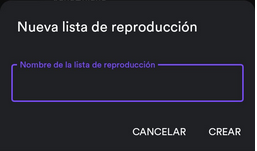


1. Reproducir siguiente: Agrega el archivo a la lista de reproducción actual.
2. Agregar a cola de reproducción: Cumple exactamente la misma función anteriormente mencionada (probablemente accidental) (¿Bug?)
3. Agregar a lista de reproducción: Permite agregar el archivo a una lista de diferente a la actual, despliega el siguiente menú:



Las opciones inferiores a “Nueva lista de reproducción” añaden el archivo a la lista de reproducción indicada.

La opción “Nueva lista de reproducción” despliega el siguiente cuadro (Text Box):

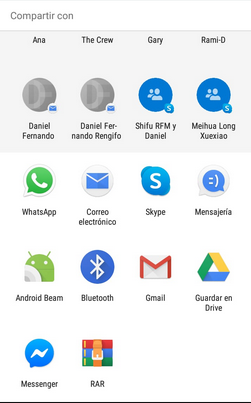


De esta forma se da la creación de

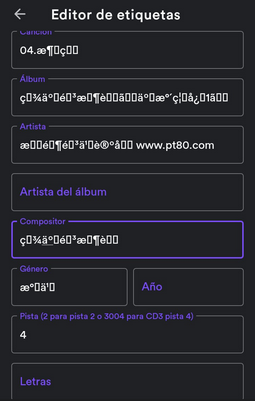
nuevas listas

de reproducción personalizadas.

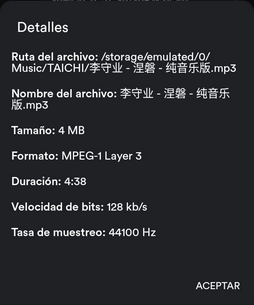
1. Ir al álbum: Abre la lista relacionada con el álbum del archivo seleccionado.
2. Ir al artista: Abre la lista relacionada con el artista del archivo seleccionado.

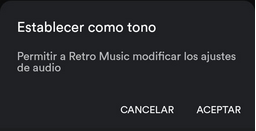


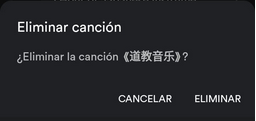
1. Compartir: Despliega el menú de redes sociales (Sharing Panels)



1. Editor de etiquetas: Despliega el formato de etiquetas (independiente de archivo). Este formato (Text Fields) sobre escribirá las etiquetas presentes en el archivo. En caso de mantenerse inéditas, se mantendrá el estado actual en la etiqueta correspondiente.



1. Detalles: Despliega la información detallada del archivo (Dialogue).
2. Establecer como tono de llamada: Despliega un dialogo de aceptación (Dialogue), en el cual se le pide permiso al usuario para cambiar sus parámetros de llamada. Al acceder la aplicación redirige al usuario al menú de configuración del dispositivo.



1. Borrar del dispositivo: Despliega un cuadro de aceptación (Dialogue), en el cual se le pide al usuario confirmación de la orden de borrado. Al confirmarla, se borrará el archivo del dispositivo.

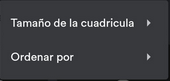
Los despliegues aquí documentados cumplen con la misma transición antes descrita entre elementos de analítica y sus respectivas listas, en el menú principal. Las excepciones se encuentran en redirecciones como la de la opción “Compartir” y “Establecer como tono de llamada”. En cada cambio al estado interno de la aplicación se utiliza una notificación de Toast:



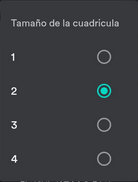
* Nota: no se hace diferenciación entre casos plurales o singulares, en una funcionalidad mayoritariamente singular, se utiliza un default plural.

Probablemente se deba a disparidades entre idiomas, puesto que Retro Music Player soporta treinta idiomas.

El icono de menú en la barra de búsqueda presente en este menú, despliega el siguiente menú:

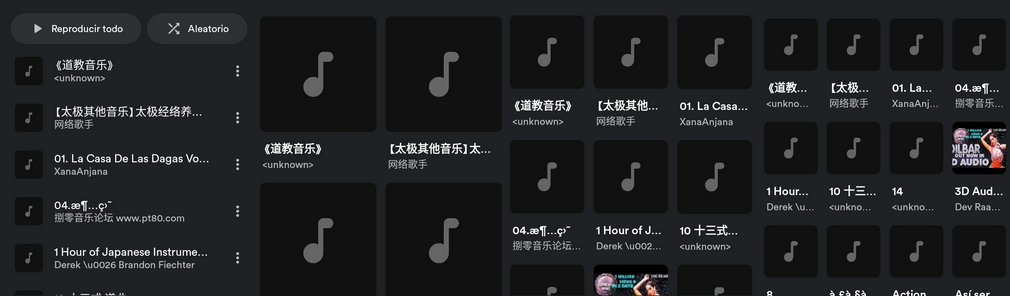


El cual se enfoca en cambios en la visualización de lista, por medio de las siguientes opciones:

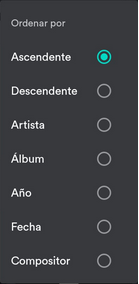


1. Tamaño de la cuadricula: Permite editar, tanto el modo de visualización entre lista (List) y cuadricula (Grid List)); como la dimensión de los componentes individuales.

1 2 3 4



* Nota: Como podemos ver, el orden de opciones no representa acertadamente la progresión dimensional. Además de esto solo la primera opción contiene el menú de reproducción. (¿Bug?)



1. Ordenar por: Despliega el siguiente cuadro de selección (Dialogue).

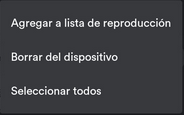
* Nota: Las opciones de criterio (Etiquetas), y las opciones de modo de ordenamiento (Ascendente/Descendente), son mutuamente excluyentes, por lo que, en cado de querer ordenar bajo ambos criterios se requiere una doble apertura del menú. Además de esto, el proceso es extremadamente contrainuititvo, atentando directamente contra la identidad de la aplicación.

Al seleccionar de forma mantenida cualquiera de los elementos, la aplicación permitirá al usuario seleccionar múltiples elementos, alterando a su vez la barra de búsqueda:



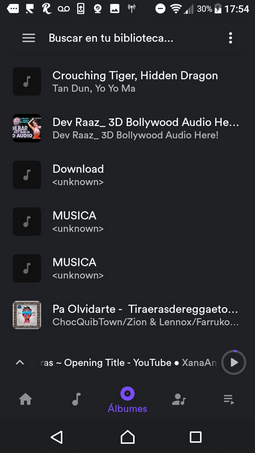
 Nueva App Bar

Menu de Opciones



En primer lugar, se informa al ususario por medio de texto, cuantos elementos tiene sleeccionados. EL icono de flecha añade los archivos seleccionados (en el orden en el que fueron seleccionados) a la cola de reproduccion. El siguiente icono cumple la misma funcion, sin cambios aparentes (¿Bug?). Finalmente el icono de opciones despliega el menú arriba representado. Las primeras dos funciones cumplen las mismas funciones antes descritas, mas hacen referencia a la selección completa. La ultima opcion permite sleccionar todods los elementos de la vizualicacion presente.

1. Listas de Etiquetas:

-------------------- ----------------------------- Estático Barra de Búsqueda III -------------------- ----------------------------- Scroll Lista - - - - - - - - - - - - - - - - - - - - - - - - ------------------------------------------------- Estático Reproducción Actual (opcional) - ------------------------------ Barra de Tareas -------------------------------------------------

La visualización de lista (List), se utiliza para representar colecciones de archivos de audio relacionados con una cierta etiqueta, específicamente “álbum” y “artista”. A diferencia de la lista de archivos, no contiene el menú de reproducción.

Cada elemento incluye una imagen relacionada con la etiqueta, así como su nombre y otras etiquetas pertinentes.

El icono de la barra de búsqueda despliega mismo menú, descrito en el literal anterior, sin embargo, los criterios de ordenamiento se reducen a un número menor de etiquetas, específicamente año y artista/álbum, dependiendo de en qué pestaña se encuentre en el momento. Al seleccionar de manera continua cualquiera de los elementos presentes, la aplicación dará la misma respuesta descrita en la anterior visualización, en referencia a todos los elementos contenidos dentro del elemento.

Al seleccionar un elemento de la lista presente, se desplegará una de las siguientes dos vistas:

1. Visualización Etiqueta (Álbum):

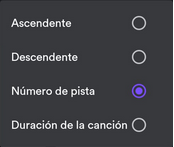


-------------------- ----------------------------- Estático App Bar I -------------------- ----------------------------- Scroll Imagen - - - - - - - - - - - - - - - - - - - ------------------------------ - Detalles - - - ------------------------------ - Menú de Reproducción - - - ------------------------------ - Lista - - - - - - --------------------------------------------------- Estático Reproducción Actual (opcional) --------------------- ------------------------------

La visualización de álbum, despliega la información relacionada con un álbum especifica. Se hace un énfasis especial hacia la imagen principal de la etiqueta, con la lista de archivos relacionados en la parte inferior de la pantalla.

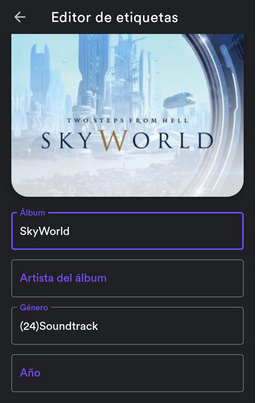
Se incluye el menú de reproducción y el menú de archivo antes ya detallados.

Finalmente, en el App Bar, se encuentran una serie de opciones:

1. 



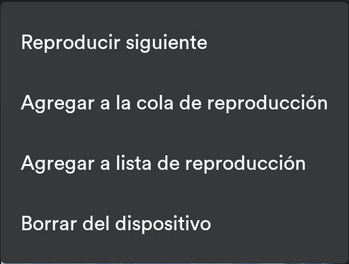
Se despliegan opciones para el ordenamiento de los archivos.

1. 



Se despliegan el formato para editar las etiquetas de la etiqueta.

* Nota: no se puede editar la imagen de la etiqueta desde esta vista.

 c.

Se despliegan las siguientes opciones, similares a sus correspondientes análogas en el menú de archivo antes descrito, más refiriéndose a la colección entera de archivos.

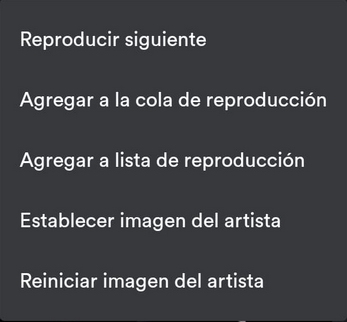
1. Visualización Etiqueta (Artista):



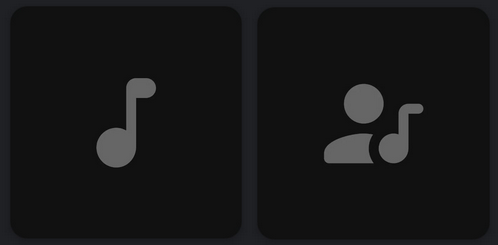
-------------------- ----------------------------- Estático App Bar II -------------------- ----------------------------- Scroll Imagen - - - - - - - - - - - - - - - - - - - ------------------------------ - Detalles I - - - ------------------------------ - Menú de Reproducción - - - ------------------------------ - Detalles II - - - - - - - ------------------------------ - Álbumes - - - - - - - - - - - - - ------------------------------ - Lista - - --------------------------------------------------- Estático Reproducción Actual (opcional) --------------------- ------------------------------

La visualización de artista, integra el despliegue de álbumes, dada su jerarquía. A la visualización de álbum se le añade una descripción bajo “Biografía” y una lista de álbumes relacionados con el artista.

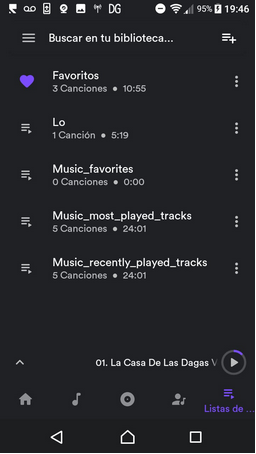
El menú del App Bar, también cambia, dejando solo uno de los iconos, cuyas funciones también difieren frente a su contraparte en la visualización de álbum:



1. Se despliegan las siguientes opciones, idénticas a sus contrapartes antes descritas. Más se añaden opciones para alterar la imagen del artista. Al seleccionar “Reiniciar imagen del artista” se reestablecerá la imagen de fábrica. Por otro lado, seleccionar “Establecer imagen del artista” redirigirá al usuario hacia el menú de selección de imagen del dispositivo

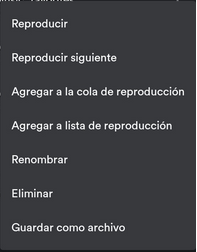
* Nota: Las opciones de cambio de imagen no se encuentran en la visualización de álbum, aun si la función sea igual de valida e importante.
* Nota: Si bien la imagen de fábrica del usuario fue consideraba como “Loable”, la imagen de fábrica para el resto de las imágenes en la aplicación sigue el mismo témplate de otras aplicaciones. A este respecto consideramos que la consistencia debe ser algo a tomar en cuenta a futuro.

1. Listas de Reproducción:

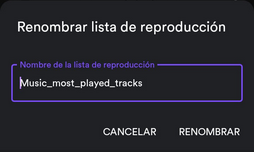


-------------------- ----------------------------- Estático Barra de Búsqueda III -------------------- ----------------------------- Scroll Favoritos - ----------------------------- - Lista - - - - - - - - - - - - - - - - - - - - ------------------------------------------------- Estático Reproducción Actual (opcional) - ------------------------------ Barra de Tareas -------------------------------------------------

La visualización de listas de reproducción presenta la misma visualización antes desértica, enfocándose en las listas de reproducción presentes en la aplicación. Al seleccionar el icono a la derecha de la barra de búsqueda, se desplegará el menú de creación de listas de reproducción antes presentado. Por otro lado, al seleccionar el icono de opciones, presente en cada elemento. Se desplegará el siguiente menú:



1. Reproducir: Esta opción, no ejecuta ninguna acción (¿Bug?)
2. Las siguientes tres opciones cumplen las mismas funciones antes discutidas, para sus correspondientes análogas en otras visualizaciones.

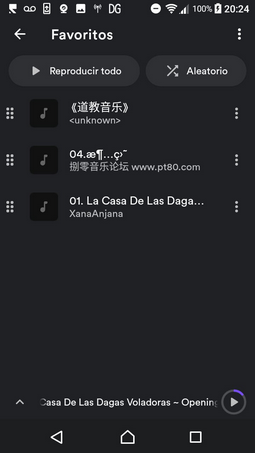


1. Renombrar: permite renombrar una lista de reproducción, por medio del siguiente cuadro (Text Field):

1. Eliminar: Permite, por medio del mismo dialogo antes descrito para funciones de eliminación, eliminar la lista de reproducción, sin embargo, no elimina los archivos relacionados.
2. Guardar como archivo: Permite guardar, en una carpeta predeterminada, la información de una lista de reproducción.

Al seleccionar uno de los elementos, se desplegará la siguiente vista:

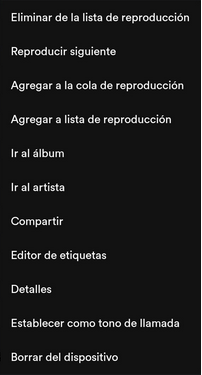
1. Lista de Reproducción:



-------------------- ----------------------------- Estático Barra de Búsqueda III -------------------- ----------------------------- Scroll Menú de Reproducción - ----------------------------- - Lista - - - - - - - - - - - - - - - - - - - - - - - - -------------------------------------------------- Estático Reproducción Actual (opcional) --------------------- -----------------------------

Esta visualización, permite representar la información contenida en una lista de reproducción, así como acceder a sus funcionalidades, por medio de seleccionar cualquier archivo de audio representado, o de usar el Menú de Reproducción. Al seleccionar el icono a la derecha del App Bar, se desplegará el menú de lista de reproducción, presentado en el literal anterior.

Por otro lado, al seleccionar el icono de opciones en cualquiera de los elementos de la lista, se desplegará el siguiente menú:

1. Eliminar de la lista de reproducción: Por medio del dialogo de aceptación antes descrito para opciones de eliminación se da la opción de eliminar el archivo de audio de la lista de reproducción, el archivo, sin embargo, no desaparecerá de la biblioteca general de la aplicación, solo de la lista de reproducción correspondiente.
2. Las siguientes tres opciones son idénticas a sus contrapartes antes descritas.
3. Ir al álbum: se redirige al usuario a la vista de álbum correspondiente a la etiqueta del archivo.
4. Ir al artista: Se redirige al usuario a la vista de artista correspondiente a la etiqueta del archivo.
5. Las siguientes opciones son idénticas a sus contrapartes antes descritas.

* Nota: Esta vista será utilizada por las listas de archivos extraídas de la micro-analítica presente en el menú principal de la aplicación.

1. Lista de Reproducción:

-------------------- ----------------------------- Estático Imagen - - - - - - - - - - - - - - - - - ------------------------------ Progreso - ----------------------------- - Detalles - ----------------------------- - Controles - - - ------------------------------ Opciones -------------------------------------------------

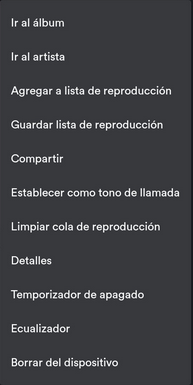
Esta visualización presenta la información disponible sobre la reproducción actual. Emerge tras seleccionar el icono a la izquierda de la “reproducción actual” en cada vista antes discutida.

Una vez más se hace un gran enfoque visual sobre la imagen ligada a la reproducción. Además de lo cual se incluye una barra de progreso modificable.

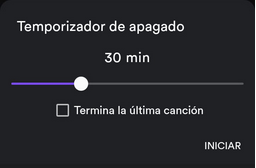
Los controles presentan cinco funcionalidades esperadas de cualquier reproductor de audio, de forma intuitiva. Permitiéndole al usuario alterar la forma como se reproducen los archivos (aleatorio, bucle) así como cambiar de archivo, moviéndose sobre la cola de reproducción actual. Finalmente, de manera predominante se observa el botón que da la funcionalidad de pausa.

Finalmente se tiene el menú de opciones, el cual comienza desde la izquierda con el icono invertido, cuya función es la de volver a la vista anterior, desde la cual se accedió a la actual. El corazón permite añadir el archivo a la lista de reproducción por defecto “Favoritos”.

La siguiente opción despliega la vista de cola de reproducción, que discutiremos más adelante. Por último, el icono de opciones despliega el siguiente menú:



1. las primeras ocho opciones son idénticas a sus contrapartes correspondientes.



1. Temporizador de Apagado: Permite determinar un temporizador para apagar el dispositivo, a través del siguiente dialogo:

* Nota: La opción “Termina la última canción” permite terminar de reproducir el ultimo archivo, una vez se acabe el tiempo estipulado en el temporizador, antes de apagar el dispositivo.

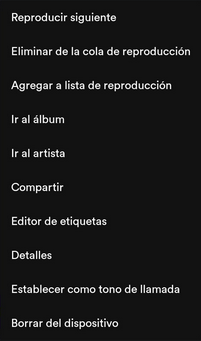
1. Ecualizador: Redirige al usuario hacia el menú de configuración de output de sonido del dispositivo.
2. La última opción es idéntica a su contraparte antes descrita.
3. Cola de Reproducción:



-------------------- ----------------------------- Scroll Informacion - - - ----------------------------- - Lista - - - - - - - - - - - - - - - - - - - - - - - - - ----------------------------- - Botón de Limpieza -------------------- -----------------------------

Esta visualización presenta la cola de reproducción, la cual es la fuente directa de las reproducciones que se llevan a cabo en la aplicación. El índice al lado izquierdo de cada elemento es su índice dentro de esta cola, aquellos inferiores a 0 son aquellos ya reproducidos en esta iteración de la cola.

Al seleccionar el icono de opciones al lado derecho de cada elemento se desplegará el siguiente menú:

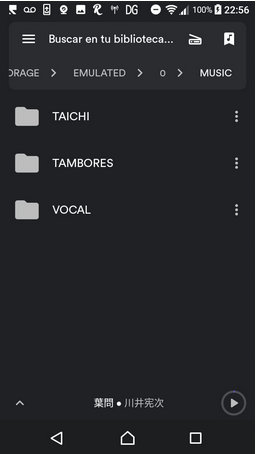


1. la primera opción es idéntica a su contraparte antes descrita.
2. Eliminar de la cola de reproducción: Elimina el registro del archivo, de la cola de reproducción, a través del dialogo antes descrito para opciones de eliminación.
3. Las siguientes opciones son idénticas a sus contrapartes antes descritas.

Finalmente, el botón “Limpiar Cola de Reproducción”, permite eliminar todos los registros de la cola de reproducción.

Nota: Si este botón es seleccionado, antes de que un proceso de scroll haya culminado, la app se cerrara abruptamente. (¡BUG!)

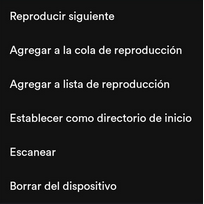
1. Carpetas:



-------------------- ----------------------------- Estático Barra de Busqueda - ----------------------------- - Navegación -------------------- ----------------------------- - Lista - - - - - - - - - - - - - - - - - - - - - - -------------------- ----------------------------- Estático Reproducción Actual (opcional) -------------------- -----------------------------

La visualización de carpetas permite navegar todo el almacenamiento local del dispositivo, buscando archivos de audio para incluirles en la biblioteca de la aplicación.

Al seleccionar el icono de opciones a la derecha de cada elemento en la lista, se desplegará el siguiente menú:



1. Las primeras tres opciones son idénticas a sus contrapartes antes descritas.
2. Establecer como directorio de inicio: Establece la carpeta seleccionada, como carpeta principal de la biblioteca de la aplicación, es decir la carpeta que se analizar automáticamente a la hora de construir la biblioteca.
3. Escanear: Se analiza la carpeta seleccionada, añadiendo los archivos de audio encontrados a la biblioteca de la aplicación.
4. La última opción es idéntica a sus contrapartes correspondientes.

Finalmente, las opciones en el App Bar incluyen las siguientes funcionalidades:



1. Permite escanear automáticamente el directorio actual.



1. Regresa automáticamente al directorio de inicio.
2. Configuración:



-------------------- ----------------------------- Estático App Bar -------------------- ----------------------------- Scroll Versión Pro - - - - - - - - - ----------------------------- - Lista - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -------------------- -----------------------------

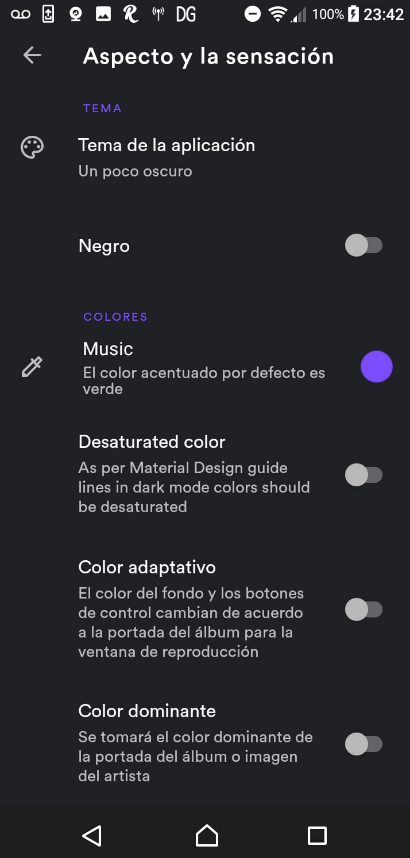
La visualización de configuración, es la visualización con colores e iconos más variados. La aplicación depende íntimamente de la estética intrínseca de las imágenes de artistas, álbumes y archivos presentes en su biblioteca.

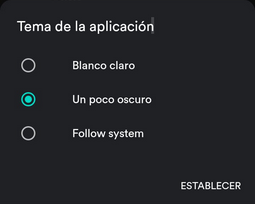
* 1. Versión Pro:



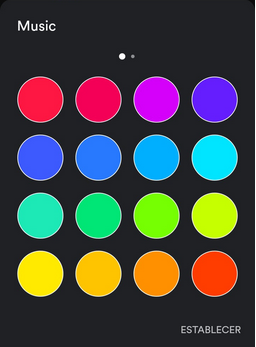
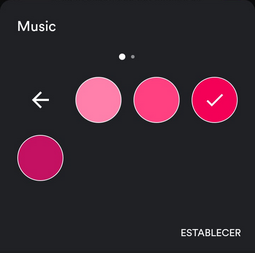
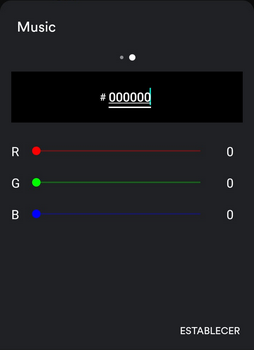
La versión pro, informa al usuario de los beneficios obtenidos al comprar la versión.

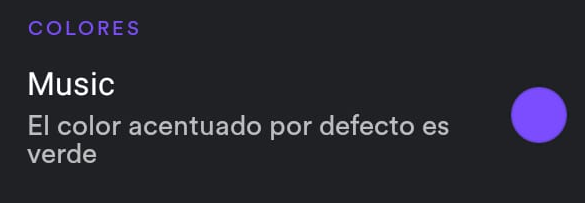
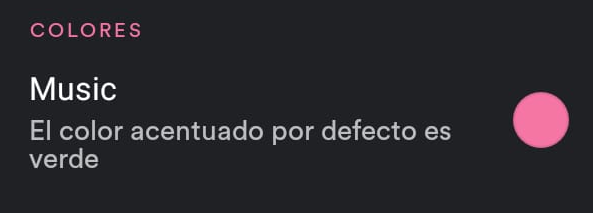
Los botones presentes operan bajo las funcionalidades de pago y restauración ofrecidas por Google Play.

* 1. Aspecto y la Sensación:
     1. Tema de la aplicación: Permite seleccionar el esquema de colores de la aplicación:

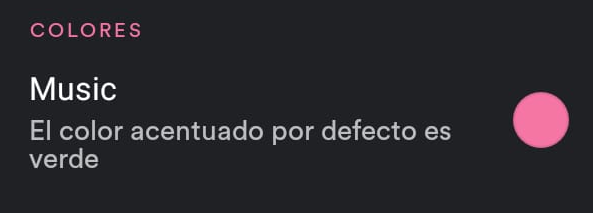
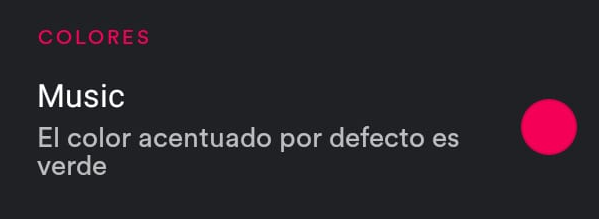


* + 1. Negro: (Solo versión pro) permite invertir los colores de la aplicación. (Funcionalidad que existe por defecto en Android).
    2. Music: Permite seleccionar el color default de la aplicación por medio de los siguientes menús de selección:

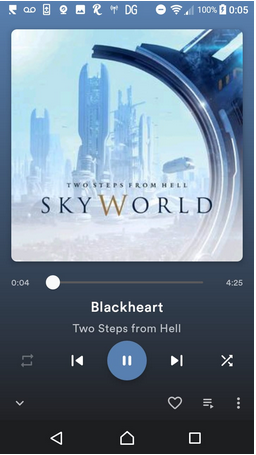
  

* Nota: El título “Music”, se encuentra en inglés, mientras el resto de la aplicación se encuentra escrita en español. (¡Bug!)
  + 1. Desaturated color: Ofrece la opción de desaturar el color, siguiendo los lineamientos de Material Design.
* Nota: Esta opción y su descripción se encuentran en inglés, mientras que el resto de la aplicación esta en español. (¡Bug!)

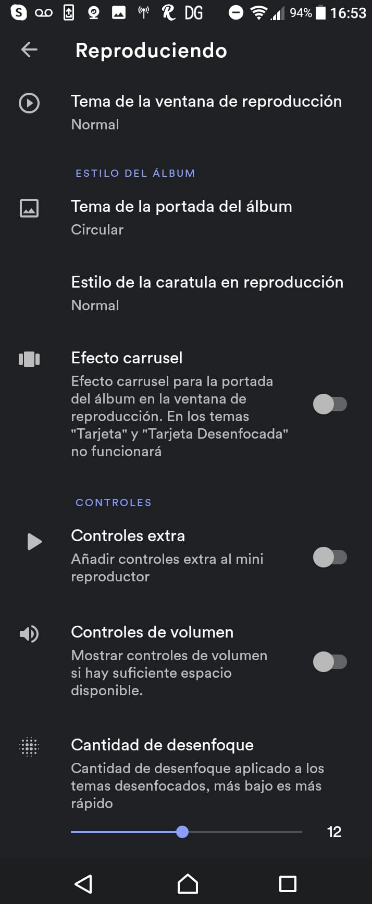
 

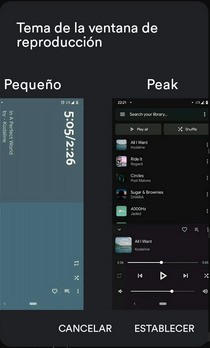
* + 1. Color adaptativo: Ofrece la opción de adaptar el color del fondo de pantalla y los botones de la aplicación de acuerdo con el color dominante encontrado en la imagen ligada a la reproducción actual.

* + 1. Color Dominante: Ofrece la opción de determinar los colores predominantes a partir del color dominante de la imagen ligada a la reproducción actual.
* Nota: No se encontró diferencia sustancial entre ambos estados del parámetro. (¿BUG?)

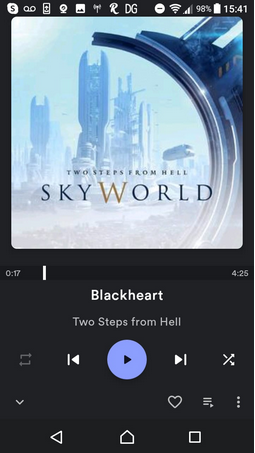
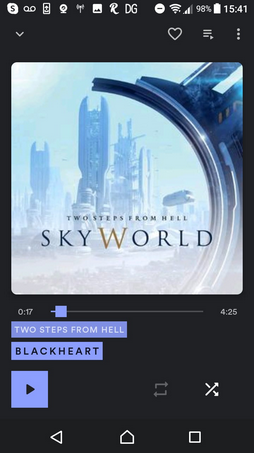
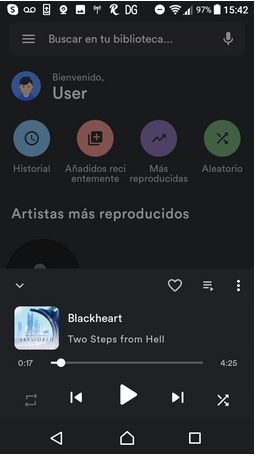
1. Reproduciendo:



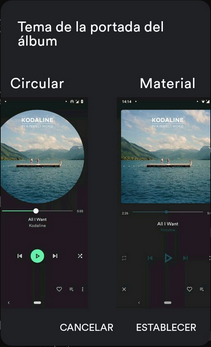


* + 1. Tema de la ventana de reproducción: permite al usuario cambiar la visualización de archivo antes presentada, manteniendo sus funcionalidades principales. Los temas disponibles para usuarios gratuitos son:

Normal Plano Fit Pequeño Peak

Nota: La representacion de menu “Fit”, difirere sustancialmente de la ejecutada. Refleja mas bien, una opcion en el la siguiente opción del menú. (¿BUG?)



* + 1. Tema de la portada del album: permite al ususario determinar la vizualizacion de la imagen relacionada con los diferentes albumes dentro de la biblioteca.

Nota: No fue posible apreciar el cambio producido por estas opciones. (¿BUG?)



* + 1. Estilo de la caratula en reproducción: Permite al usuario cambiar la forma en la cual la caratula de un archivo en reproducción es representada, por medio del siguiente menú.

Nota: Una vez más, no fue posible determinar el cambio causado al alternar entre estas opciones. (¿BUG?)

* + 1. Efecto carrusel: Exclusivamente para la versión pro, permite al usuario iniciar un efecto carrusel entre las diferentes reproducciones mientras estas son reproducidas.
    2. Controles extra: permite incluir botones de reproducción a la versión minimizada de la visualización de archivo. Como se aprecia a continuación:

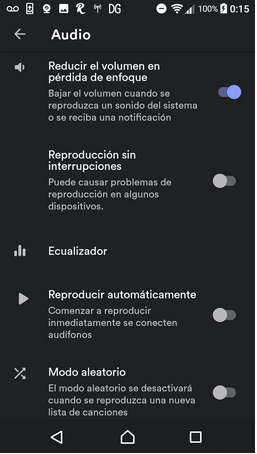
 

* + 1. Controles de volumen: Permite incluir un controlador de volumen a la visualización de archivo.

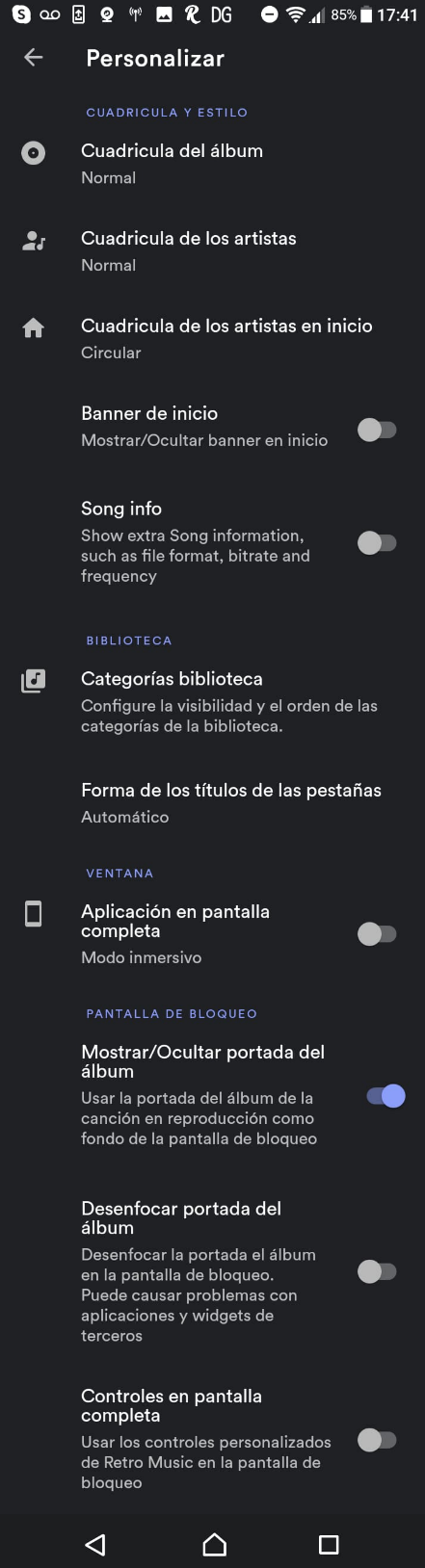
* + 1. Cantidad de desenfoque: Permite, para ciertos temas unciamente disponibles en la version pro, incrementar o siminuir el nivel de desenfoque aplicado.

1. Audio:

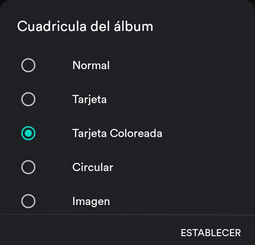


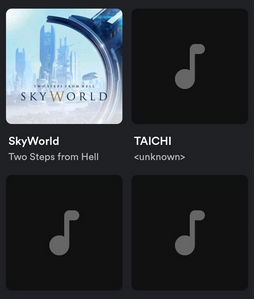
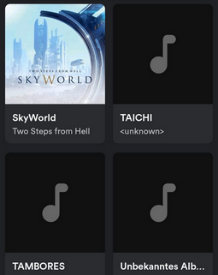
* + 1. Reducir el volumen en perdida de enfoque: Permite al usuario reducir el volumen de la reproducción, mientras le dispositivo no este centrado en la aplicación.
    2. Reproducción sin interrupciones: Permite al usuario dar prioridad exclusiva a Retro Music Player, sobre el output de sonido.
    3. Ecualizador: Redirige al usuario hacia la configuración de output de sonido del dispositivo.
    4. Reproducir automáticamente: Permite al usuario dar permiso a la aplicación para comenzar la reproducción de forma inmediata al conectar audiófonos al dispositivo, mientras exista una reproducción en cola y la aplicación este siendo enfocada por el dispersivo.
    5. Modo aleatorio: permite al usuario alterar la desactivación automática del modo aleatorio de reproducción.

1. Personalizar:



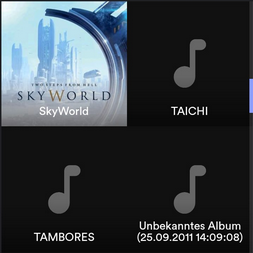
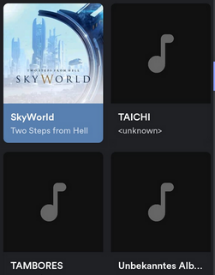
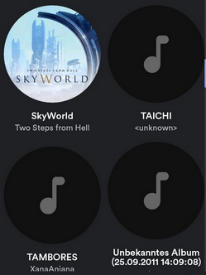
* + 1. Cuadricula del álbum: despliega el siguiente menú, el cual le permite al usuario determinar la visualización de elementos en la vista de álbumes.



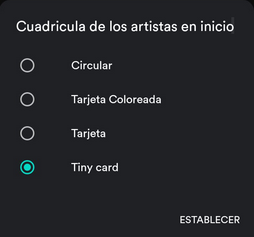


Normal Tarjeta

Tarjeta Coloreada Circular Imagen

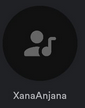


* + 1. Cuadricula de artistas: Cumple las mismas funciones presentadas en el anterior literal, para la visualización de artistas.



* + 1. Cuadricula de los artistas en inicio: Cumple las mimas funciones antes descritas, sin embargo, su afecta únicamente el menú de inicio, y las opciones difieren sutilmente:

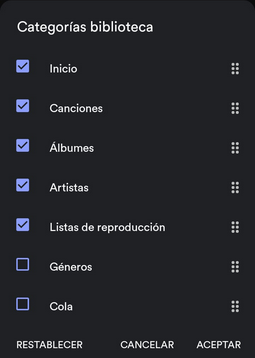
Circular Tarjeta Coloreada Tarjeta Tiny card

* Nota: La opción Tiny card, es la única que no sigue el patrón de idioma.
* Nota: Esta opción no tiene una contraparte para los álbumes presentes en el inicio.
  + 1. Banner de inicio: Permite activar o desactivar la visualización del background del perfil de usuario:

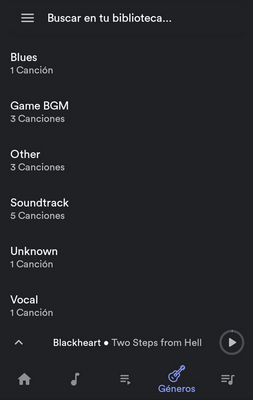
* + 1. Song Info: Permite desplegar información extra, en referencia a los archivos individuales.
* Nota: La opción Song Info, no sigue el patrón de idioma determinado.
* Nota: No fue posible determinar el cambio causado por esta funcionalidad.



* + 1. Categorías biblioteca: Despliega el siguiente menú, dándole al usuario la opción de intercambiar dos elementos de su barra de navegación, por otras alternativas menos desarrolladas (Géneros y Cola).



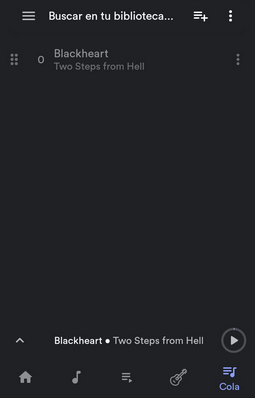
* + Género: Despliega la etiqueta Genero, usando la siguiente visualización, donde cada elemento es un género. No se incluyen más funcionalidades, al no ser el enfoque del desarrollo.



Cada elemento contiene una colección de archivos relacionados con la etiqueta de género. La visualización utilizada es idéntica a la de lista de reproducción antes descrita.



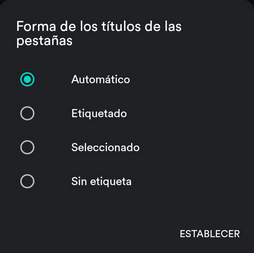
* + Cola: Despliega la visualización de cola de reproducción. Omitiendo las funcionalidades de borrado en su contraparte antes descrita:

El botón de opciones en la esquina superior derecha, no cumple ninguna función. Aun si despliega un menú con la opción “Ordenar por” (el cual simplemente será iluminado al ser seleccionado, sin activar funcionalidad alguna.



El botón de añadidura a la izquierda del mismo despliega el dialogo de creación de listas de reproducción. Función que consideramos profundamente inconexa.

Tampoco se cumplen con funcionalidades de selección multimple en esta vista.



* + 1. Forma de los títulos de las pestañas: Permite editar la visualización de los subtítulos en la barra de navegación, por medio del siguiente menú:



Automático



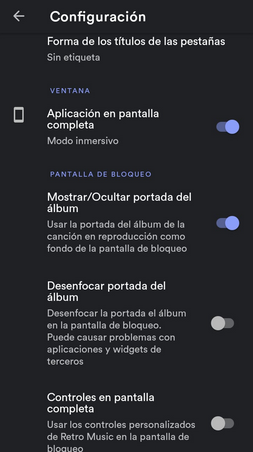
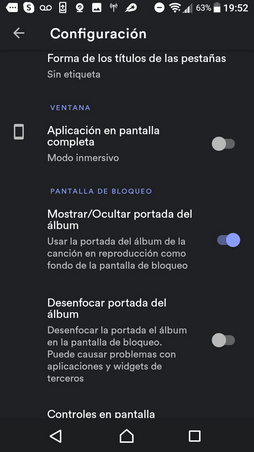
Etiquetado

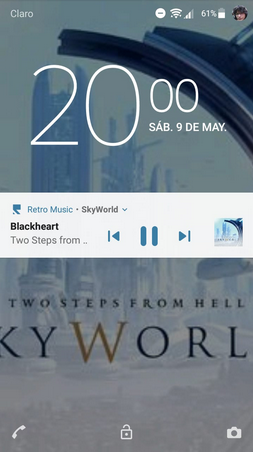
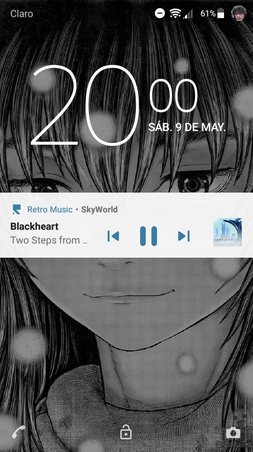


Seleccionado

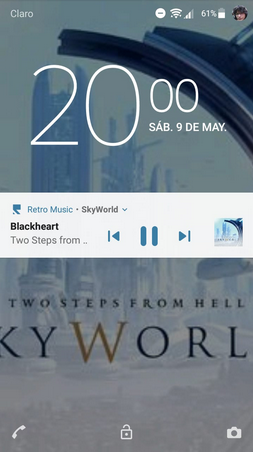
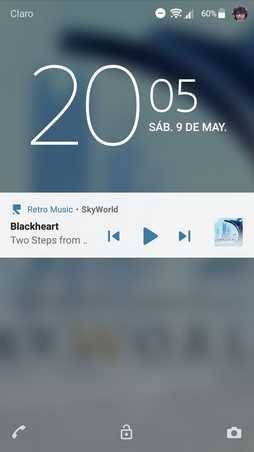


Sin etiqueta

* Nota: No se encontraros diferencias sustanciales entre la opción “Automático” y “Seleccionado”.
  + 1. Aplicación en pantalla completa: Permite al usuario desaparecer la Barra inferior (Bottom Navigation Bar) y la barra de estado (Phone Status Bar), en aras de obtener una visualización más predominante de la aplicación. 
    2. Mostrar/Ocultar portada del álbum: Permite al usuario usar la portada del álbum ligada a la reproducción actual como pantalla de bloqueo.

* + 1. Desenfocar portada del álbum: Permite al usuario usar un efecto de difuminarían sobre la anterior opción. (disponible solo si el literal anterior se encuentra positivo.

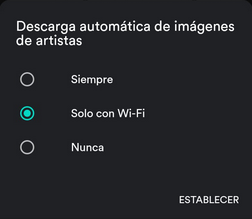
 

* + 1. Controles en pantalla completa: Permite desplegar los controles de Retro Music Player en la pantalla de bloqueo.
* Nota: Fue imposible determinar las diferencias causadas por esta opción. (¿Bug?)
* Nota: El título de la opción hace referencia a la pantalla completa, mientras que la categoría hace referencia a la pantalla de bloqueo.

1. Imágenes:

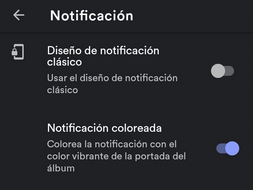


* + 1. Ignorar las portadas de la biblioteca de medios: Permite al usuario Ignorar la descarga de imágenes relacionadas con la biblioteca de medios.

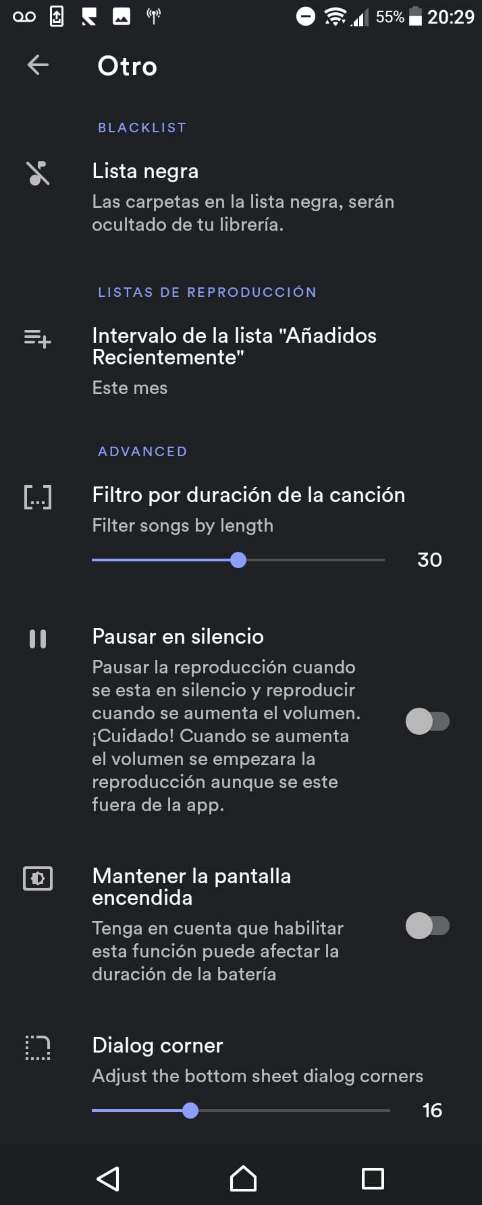


* + 1. Descarga automática de imágenes de artistas: Despliega el siguiente menú, que permite al usuario decidir los medios de descarga de las imágenes relacionadas con la biblioteca de la aplicación.

1. Notificaciones:

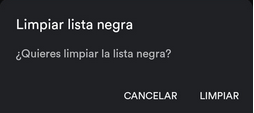


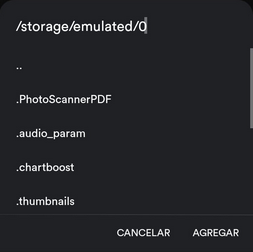
* 1. Diseño de notificación clásico: Permite al usuario cambiar el icono de notificación.
  2. Notificación coloreada: Permite al usuario determinar si el icono de notificación contiene color.
* Nota: Para ambas opciones fue imposible determinar el cambio en el icono de notificación. (¿Bug?)

1. Otro:
   * 1. Lista Negra: Despliega el siguiente menú, permitiéndole al usuario administrar carpetas de las cuales no se extraerán archivos de sonido.

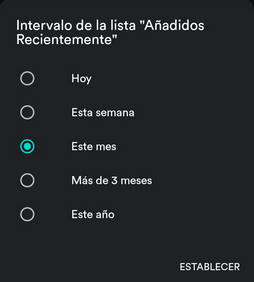


* + La opción Limpiar, permite descartar todas las carpetas de la lista, revirtiendo su estado a validas, por medio del siguiente dialogo:

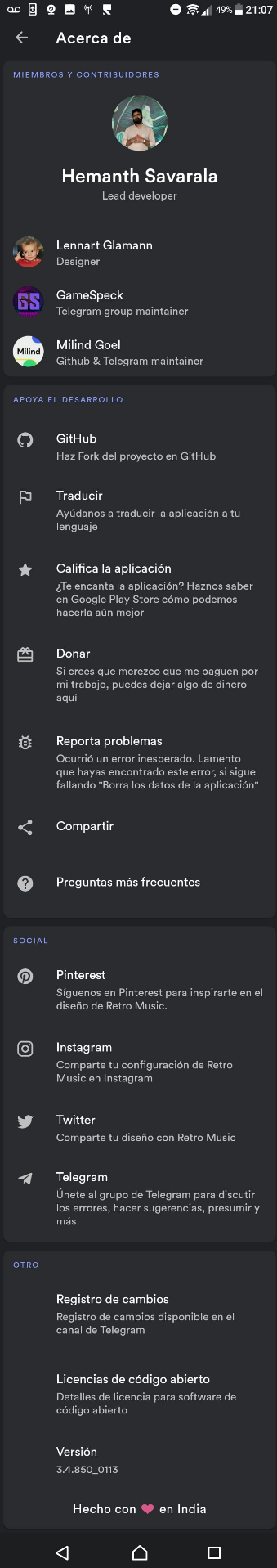


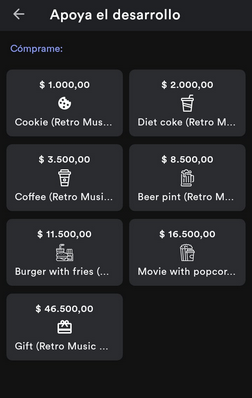


* + La opción agregar, despliega un menú poco intuitivo y difícil de manejar, en el cual se listan los directorios disponibles, permitiéndole al usuario agregar directorios a la lista negra:



* + 1. Intervalo de la lista “Añadidos Recientemente”: Permite al usuario alterar el corte temporal para la métrica “Añadidos Recientemente”, a través del siguiente menú:
    2. Filtro por duración de la canción: Permite al usuario ejercer un filtro en términos de minutos de duración, frente a la adición de archivos de audio a la biblioteca.
    3. Pausar en silencio: Permite al usuario pausar la reproducción de un archivo de audio en cuando el volumen del dispositivo sea igual a cero. Una vez el volumen pierda dicho valor la reproducción continuara automáticamente.
* Nota: Como se mencionó anteriormente, este menú se utiliza como un repositorio misceláneo. Sin embargo, ya antes de ha discutido la funcionalidad de audífonos, extremadamente similar a esta. Por esta razón consideramos que esta opción no pertenece a este menú, dado el énfasis en navegabilidad que se le da a la aplicación.
  + 1. Mantener la pantalla encendida: Permite al usuario darle prioridad a la aplicación sobre el auto-apagado de pantalla, intrínseco al dispositivo.
    2. Dialog corner: Permite al usuario alterar el tamaño de los diálogos presentes en la aplicación.
* Nota: Fue imposible determinar la diferencia incurrida al alterar este parámetro. (¿Bug?)
* Nota: Una vez más, el título de la opción no sigue los parámetros de idioma requeridos.

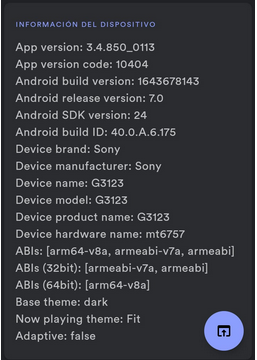
1. Acerca de:
   * 1. Miembros y Contribuidores: Cada link redirige al usuario al repositorio Github del miembro relacionado.
     2. Apoya el Desarrollo:
        + - Github: Redirige al usuario al repositorio Github de la aplicación.
          - Traducir: Redirige al usuario a link monkeycodeapp donde se traduce la aplicación.
          - Califica la aplicación: Redirige al usuario a la funcionalidad de calificación de Google Play.
          - Donar: Despliega el siguiente menú de donación:



Al seleccionar cualquier elemento, se redirige al usuario a la funcionalidad de micro-transacciones de Google Play.

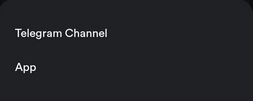


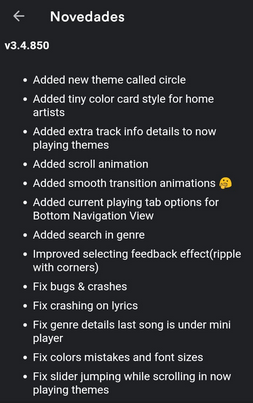
* + - * + Reporta problemas: Despliega el siguiente formato de reporte de issues, permitiendo al usuario informar de los mismos por medio del siguiente formato, el cual permite enviar el mensaje tanto a Github como al sitio web de reporte de problemas:



Además de esto, se incorpora una descripción del dispositivo del usuario:

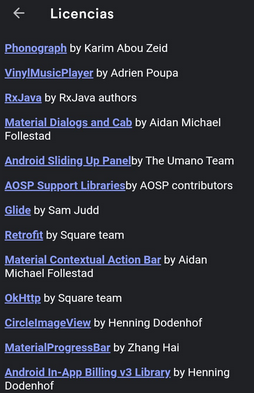
* + - * + Compartir: Redirige al usuario al menú de opciones sociales (Sharing Panels) del dispositivo.
        + Preguntas más frecuentes: Redirige al usuario a la sección FAQ en el repositorio Github.
    1. Social: Se le dan a usuario opciones, que redirigen a las diferentes redes sociales relacionadas con el desarrollo (Pinterest, Instagram, Twitter, Telegram).
    2. Otro:
       - * Registro de Cambios: Despliega la lista de cambio a través de las diferentes actualizaciones de la aplicación. La forma como dicha información es desplegada se deja a juicio del usuario por medio del siguiente menú:





La primera opción redirigirá al usuario a la sección de Telegram, que registra los cambios. Por otro lado, la segunda opción desplegara la siguiente vista:

* + - * + Licencias de código abierto: Despliega la siguiente vista, dando información e hyperlinks dirigidos a los diferentes repositorios del código abierto utilizado en la aplicación.



1. Esquema de Colores:

Como se ha observado a lo largo de este análisis visual, el esquema de colores por defecto enfatiza un fondo oscuro con elementos semi-fosforescentes. Estos principios ayudan a diferenciar elementos de interés.

El esquema estético no solo debe depender de las adiciones estéticas provistas por las imágenes relacionadas a archivos de audio, sino que además debe ser capaz de adaptarse, la aplicación provee múltiples formas de adaptar el color de las visualizaciones, más podríamos considerar que alguna forma de delimitar elementos oscuros podría ser beneficiosa para el desarrollo futuro de la aplicación.

Considerando el énfasis que se da frente a mantener al usuario alejado de parametrizaciones complejas, sin embargo, consideramos que un esquema de colores tan enfocado hacia la diferenciación es una excelente decisión de diseño.

1. Conclusiones:

Podemos observar, a lo largo de toda la aplicación un énfasis, no siempre exitosos, hacia la navegabilidad, tanto a nivel de funcionalidades como de diseño. Podemos ver dicho énfasis en la forma en la cual, a medida que la aplicación se aleja de sus funcionalidades básicas, observamos más casos de poca claridad o de problemas directo, así como funcionalidades erróneas.

Creemos, sin embargo, que la forma en la cual la aplicación se está llevando a futuro, implicara más carga de este tipo de funcionalidades poco exitosas. Hemos observado un deterioro paulatino en la confiabilidad de las funciones y la navegabilidad a medida que avanzábamos hacia los menús menos abiertos en la aplicación, observando funcionalidades poco instrumentales para la existencia de la aplicación.

Consideramos que muchas de estas actualizaciones han sido innecesarias a gran escala, dada la identidad de Retro Music Player (La que expone para sí misma, al menos). Sin embargo, consideramos que este camino no es del todo errado, puesto que, a toda probabilidad, estas funcionalidades han emergido como respuesta a retroalimentación e usuarios, y la localización de estas funcionalidades esta extremadamente lejana de las actividades y vistas más frecuentes en la aplicación, por lo que podría mantenerse un equilibro de funcionalidades a mediano o incluso largo plazo.

* Prototipado:

Si bien, no se tiene acceso al prototipo inicial de la aplicación, es de notar que Google Play ofrece funcionalidades beta, en la cual un usuario será capaz de instalar versiones aun no oficiales de la aplicación a fin de probar sus funcionalidades y estabilidad.

* Estructura Interna:

Como hemos visto anteriormente, esta aplicación no comprende un uso sustancial de los recursos del dispositivo, incluyendo sensores, almacenamiento, etc. A este respecto, el nivel al cual las funcionalidades de Retro Music Player se encuentran resulta ser bastante lógico.

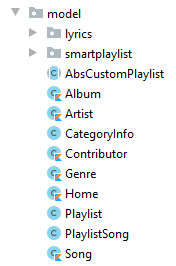
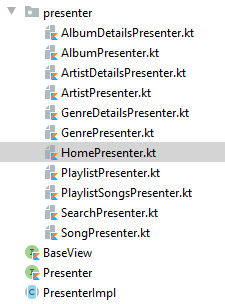
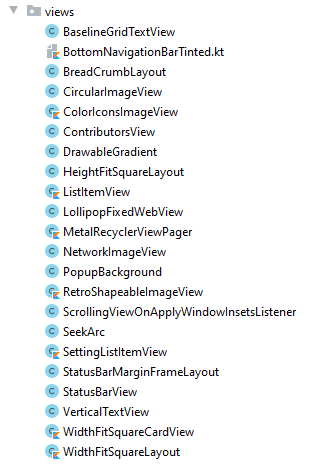
La mayoría de sus funciones son auto contendías, limitándose a el estado interno de almacenamiento y procesamiento de la aplicación. Por otro lado, la función principal, es decir la reproducción, requiere acceso a funcionalidades específicas. En momentos esporádicos la aplicación podrá requerir funcionalidades diferentes, como conexión a internet o análisis de imágenes, pero en gran medida podríamos decir que esta aplicación es bastante superficial.

Una función que llama la atención, sin embargo, es la función de escaneo la cual puede ejecutarse desde la aplicación sobre la carpeta “ROOT” sin permisos adicionales, lo cual puede requerir garantías a más bajo nivel en el dispositivo, específicamente a nivel de seguridad.

Finalmente, esta aplicación también controla niveles de prioridad en el uso del output de audio, así como procesos de background activos. Estas características también aluden a funcionalidades más profundas en el Stack de Android, cuyo estado procederemos a analizar:

* Arquitectura:

Esta aplicación usa el estilo de arquitectura clásico MVP. A ese respecto, dado su relativa simplicidad, no requiere elementos externos de ninguna índole.

En este contexto, aplica este estilo de arquitectura de forma mas bien rutinaria y sencilla. La impelementacion rodea estos tres elementos, dandole a la aplicación unn alto nivel de encapsulamiento. Esta es una excelente practica a la hroa de llevar a cabo un desarrollo, especialmente un desarrollo OpenSource. Con este estilo, se permite un alto nivel de escalabilidad a largo plazo, asi como un proceso de actualizacion saludable.

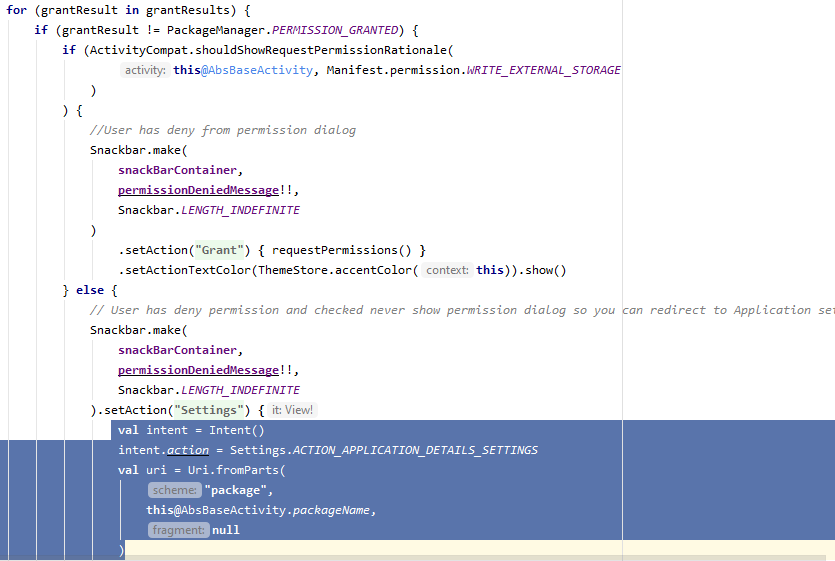
* Código:

1. Análisis de Clases:

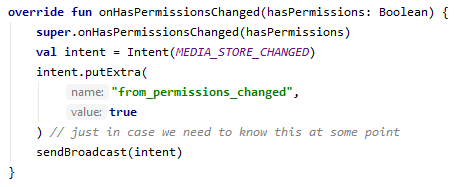
Procedernos a llevar a cabo un análisis sistemático del código de la aplicación, anotando elementos de interés como librerías, buenas y malas prácticas, micro-optimizaciones y tácticas.

Activities:

* AbsBaseActivity: Encargada primariamente del manejo de permisos, se asegura de forma redundante de confirmar los mismos. Declara variables dentro de ciclos.



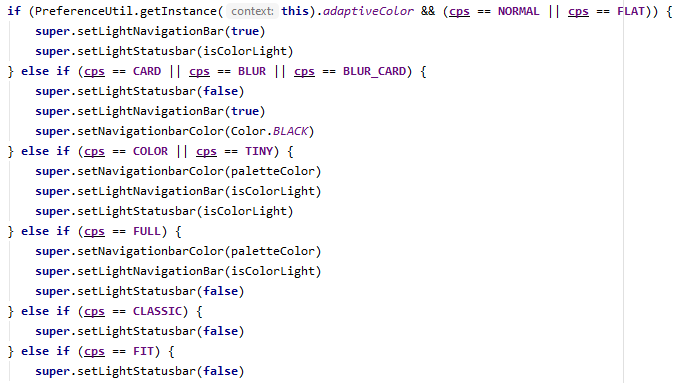
* AbsServiceMusicActivity: Se encarga del manejo de controles re reproducción. Revisa permisos de forma independiente, lo cual rompe hasta cierto punto el encapsulamiento. La granularidad de los métodos promueve una manutención sencilla.



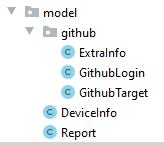
* + - * Contiene una clase: MusicStateReceiver con una referencia débil a sí misma.



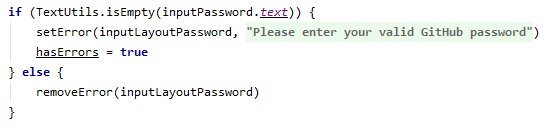
* AbsSlidingMusicPanelActivity: Maneja el panel de reproducción. Tiene malas prácticas en el manejo de condicionales, excesivamente largos y redundantes.



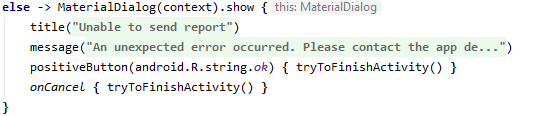
* BugReport: Maneja el reporte de problemas de forma encapsulada, cada clase en esta carpeta tiene una función clara y atómica. Incluye integración directa con GitHub.
  + - * El modelo de este elemento no está contenido en la carpeta modelos, rompiendo principios de encapsualemiento.



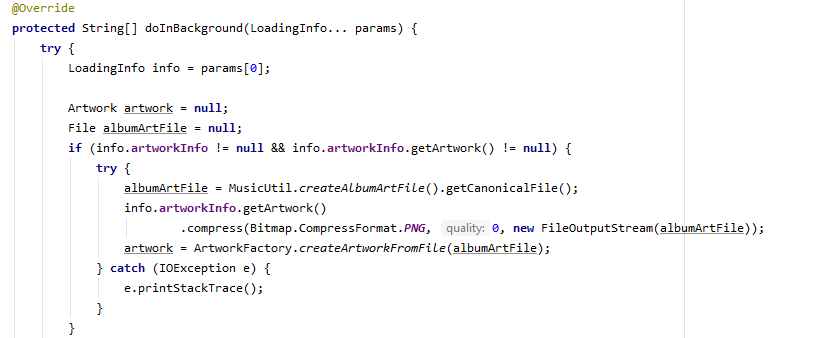
* + - * En caso de optar por GitHub, la aplicación accede y transmite las credenciales del usuario sin encriptación.



* + - * Conectividad eventual: Si se desea enviar un reporte sin internet, la aplicación lo intentara enviar. Al no ser capaz informara al usuario de un error y le pedirá que se contacte con el desarrollador. Consideramos una aproximación pobre y predispuesta a malinterpretación.

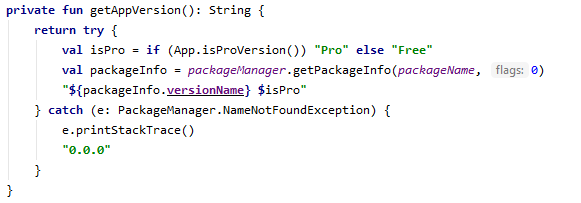


* TagEditor: Carpeta encargada del proceso de edición de etiquetas. Maneja de forma atomizada cada elemento por separado, así como cada visualización, tiene además una implementación asincrónica.
  + - * Background: Prosigue con la validación de los datos, e intenta descargar imágenes relacionadas a los archivos. Ante conectividad eventual, simplemente evita bloquearse, pero no da notificación de ningún tipo. Posteriormente se volverá a intentar el procedimiento.



* AboutActivity: Maneja el despliegue de la visualización de “About”, manteniendo el excelente nivel de encapsulamiento.

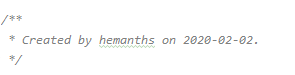
* + - * Detectamos una validación de la versión de la aplicación, la cual puede ser cambiada manualmente de “Free” a “Pro”, sin inconveniente.



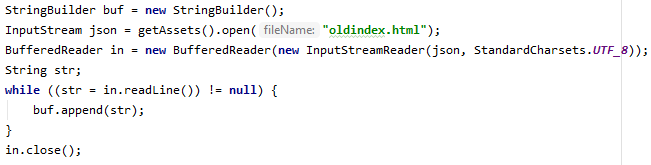
* AlbumDetailsActivity: Maneja el despliegue de la visualización de “Album”.
  + - * Utiliza preferencias para almacenar la parametrización localmente.



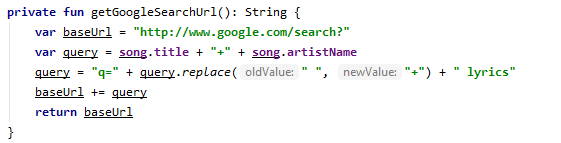
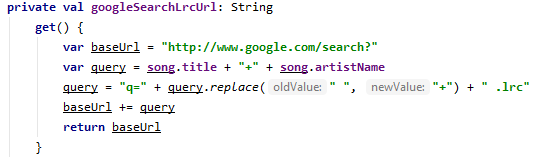
* DriveModeActivity: Maneja el modo conductor.
  + - * No se encontró la implementación de esta funcionalidad en los menús, por lo que en esencia es código muerto. Bien podría ser una actualización aun no desplegada, dado que la documentación indica la inclusión de esta actividad en febrero de este año.



* LicenseActivity: Maneja el despliegue de la visualización de licencias.
  + - * Utiliza un File, para almacenar localmente esta información.



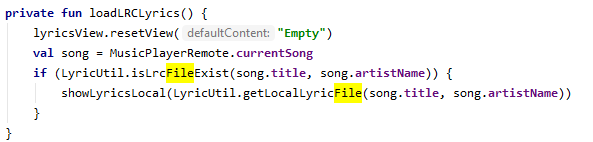
* LyricsActivity: Maneja el despliegue de la letra de las canciones. Vemos una vez más muestras de código redundante y confuso, probablemente indicación de procesos de desarrollo desordenado. Hay variables con valores duplicados, y métodos idénticos.

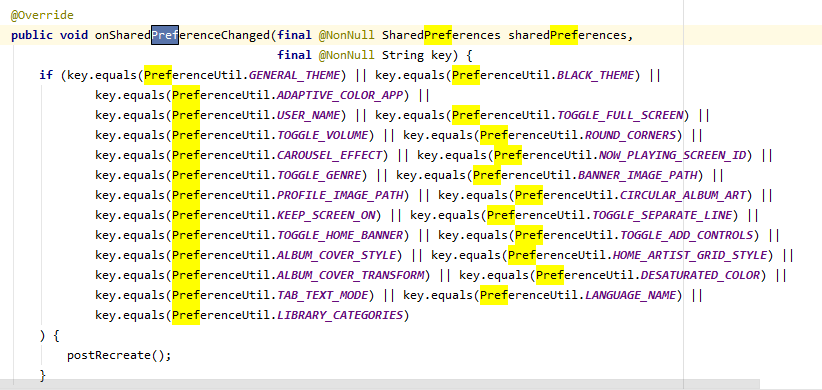
* + - * Conectividad eventual: Sigue el mismo proceso que para las imágenes.



* + - * Almacenamiento de letras en File.



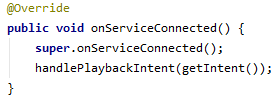
* MainActivity: Maneja los procesos de inicialización y actualización de la aplicación. Además de su estado interno de SharedPreferences.



* + - * Utiliza un Factory para obtener verificadores de actualización.



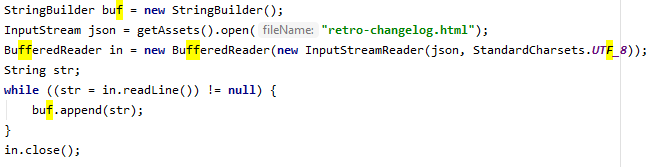
* + - * No tiene manejo de conectividad eventual, fuera de no bloquearse.



* PurchaseActivity: Maneja la compra de la versión pro, así como la restauración de copras anteriores. El procesamiento interno esta encapsulado en otras clases.
* SearchActivity: Maneja la barra de búsqueda. Usa validación de dispositivo, a la hora de intentar utilizar el input de voz.
* ShareInstagramStory: Maneja la red social Instagram.
* SupportDevelopmentActivity: Maneja la visualización de “Apoya el Desarrollo”. También encapsula el procesamiento transaccional fuera de la clase, una excelente practica de seguridad.
* UserInfoActivity: Maneja la presentación del perfil del usuario.
  + - * Almacena la información en Shared Preferences.



* WhatsNewActivity: Maneja la visualización de “Actualizaciones”.
  + - * Almacena las actualizaciones en un File.



Adapter: Transforma y adapta la información para ser entregada a los View Holders.

* MediaEntryViewHolder: Maneja la visualización de los archivos.
  + - * Rompe los patrones antes vistos, usualmente se incorporaban los ViewHolders en la clase a la cual pertenecían.

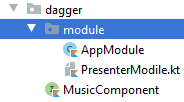


Appshortcuts: Maneja los atajos a las diferentes funcionalidades de la aplicación.

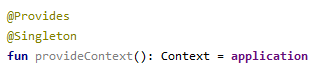
AppWidgets: Maneja los widgets encontrados en la aplicación.

Dagger:

* + - * Los nombres de clases en esta carpeta no son intuitivos, así como el nombre de la carpeta en sí. Una muy mala práctica de desarrollo, especialmente en una implementación Open Source.



* AppModule: Maneja la creación de la aplicación:
  + - * Utiliza la táctica Singleton para asegurarse de no crear más de una entidad.

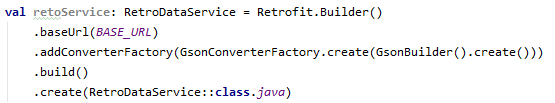


Data:

* RetroDataService: Maneja la obtención de las diferentes traducciones de la aplicación.
  + - * Usa almacenamiento externo, dirigido al repositorio de la aplicación.



* + - * Siempre descargará todas las traducciones (30)
      * No tiene manejo de conectividad eventual de ningún tipo. Se asume que, si ha logrado descargar la aplicación, tendrá conectividad para descargar el paquete de idiomas.

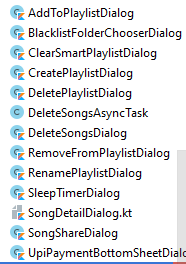


Deezer:

* DeezerApiService: Maneja la adquisición de las imágenes relacionadas.
* DeezerResponse: Encapsula la entidad de la imagen. Excelente practica de desarrollo.

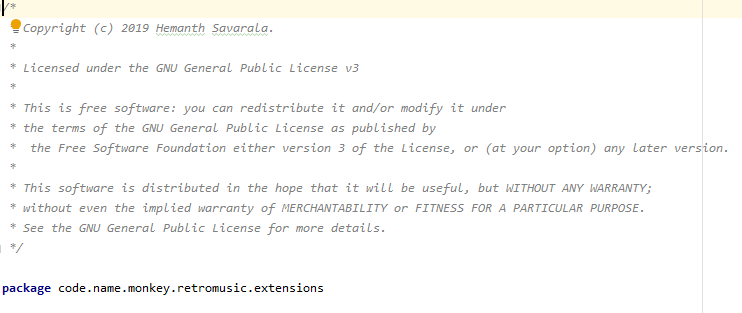
Dialogues: Maneja los diálogos de la aplicación

* + - * No se tiene la táctica de herencia antes demostrada a lo largo de la aplicación. Consideramos este punto en particular un excelente punto para usar herencia, dada la similitud de comportamiento entre diálogos.



Extensions:

* ActivityEx: Código Muerto.



* ColorExt: Maneja soporte a funcionalidades de color añadidas al sistema operativo de Android.

Fragments: Contiene los fragmentos de la aplicación.

Glide: Contiene la carga de archivos en referencia a la visualización de imágenes.

Helper: Contiene funciones misceláneas. Esta carpeta atenta directamente contra principios de encapsulamiento y mantenibilidad.

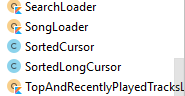
* SuperCerealScrollHelper: Código muerto.



Interfaces: Contiene las interfaces de la aplicación.

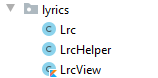
Loaders: Contiene las funciones dedicadas a extraer la información de su respectivo componente.

* + - * Contiene dos clases completamente inconexas: SortedCursor y LongSortedCursos.



Lyrics: Contiene todo lo relacionado con la letra de los archivos.

* + - * Se rompen todos los esquemas de encapsulamiento antes descritos, los elementos están en una carpeta aislada del resto del esquema de almacenamiento de código.



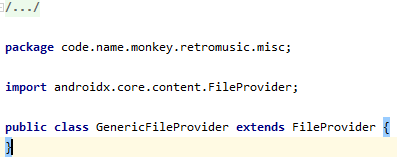
* Lrc: Encapsula la entidad de las letras.
* LrcHelper: Encapsula las funcionalidades de las letras.
* LrcView: Encapsula la visualización de las letras.

Misc:

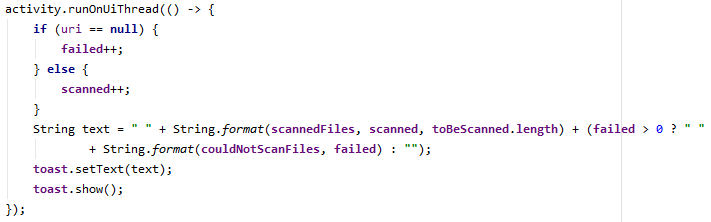
* + - * Esta carpeta, por naturaleza, quebranta los planteamientos de encapsulamiento. Los elementos en su interior son inconexos.
* AsyncSearchResultLoader: Maneja la búsqueda en background.
  + - * Utiliza AsyncTask para procesos en paralelo.



* GenericFileProvider: Código Muerto.

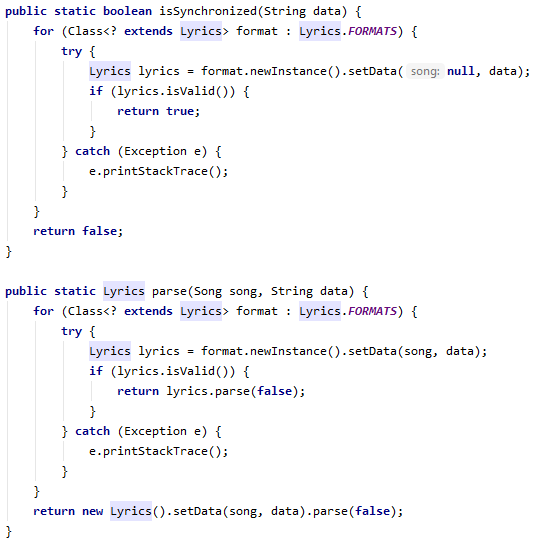


* UpdateToastMediaScannerCompletionListener: Maneja el dialogo de escaneo de la aplicación.
  + - * Utiliza runOnUiThread, para actualizar la UI en paralelo.



Model: Contiene los modelos de datos para cada elemento en la aplicación.

* Lyrics: Despliega la letra de una canción, encapsula el despliegue y la sincronización.
  + - * Valida la letra nuevamente.



MVP: Contiene los elementos del presentador, encargado de desplegar la visualización de cada elemento.

Preferences: Se encarga de almacenar y administrar los elementos de la configuración.

Providers: Contiene la implementación del almacenamiento y despliegue de la analítica, tanto miro como macro.

* + - * Cuenta con una interfaz a modo de *facade*.

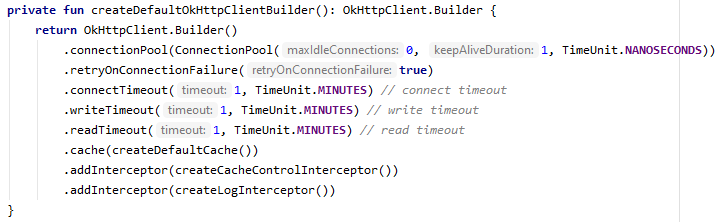


Rest:

* LastFmService: Se encarga de entablar conexión HTTP con audioscrobbles.
  + - * La llave privada de los servicios de audioscrobbles está registrada en el código como un simple String.



* LastFmClient: Encargada de obtener datos sobre los elementos de la analítica.
  + - * No posee elementos de conectividad eventual, fuera de evitar bloqueos.



* + - * Utiliza una Factory de convertidores Gson.



Room:

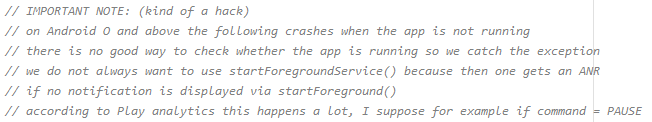
* PlaylistSongDao: Funciona como DAO para el elemento playlist.

Service:

* PlayingNotification: Despliega notificaciones de la aplicación.
  + - * Verifica versión de Android en el dispositivo.



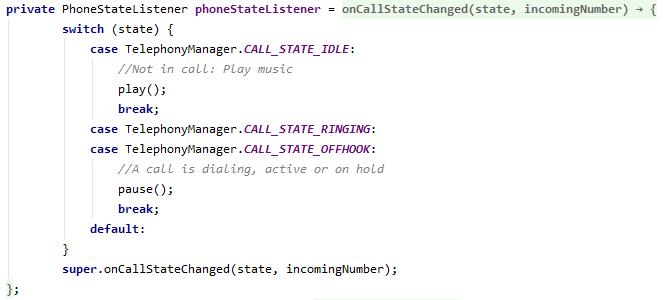
* MediaButtonIntentReceivver: Encargada de recibir y procesar los intents en referencia a la reproducción de archivos:
  + - * Se encuentran comentarios poco formales y prometedores.



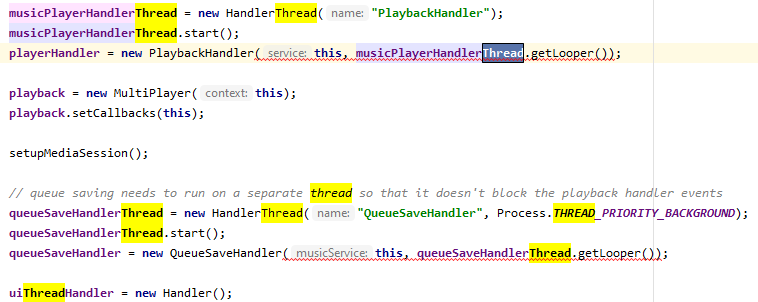
* + - * Existe un HanlderLeak en la clase.



* MusicService: Encargada de ejecutar la reproducción de los archivos.
  + - * Utiliza validación de estado.

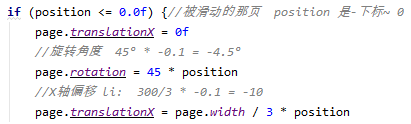


* + - * Utiliza Threads personalizados para llevar a cabo procesos en paralelo.



Transform: Contiene el manejo de transformaciones gráficas, intrínsecas al dispositivo.

* CascadingPageTransformer: Redefine el tamaño de las actividades en display.
  + - * La documentación esta escrita en un idioma asiático que no pudimos identificar, mucho menos leer. (Se informa que el autor el el desarrollador principal)





* ParallaxPageTransformer: Determina la paralaje en referencia al resto del dispositivo de las actividades de la aplicación.
  + - * Usa verificación de versión de Android del dispositivo.

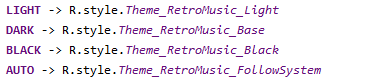


Utils: Contiene el corazón de funcionalidades de la aplicación, incluyendo las clases que incorporan la lógica y el procesamiento de los datos brutos.

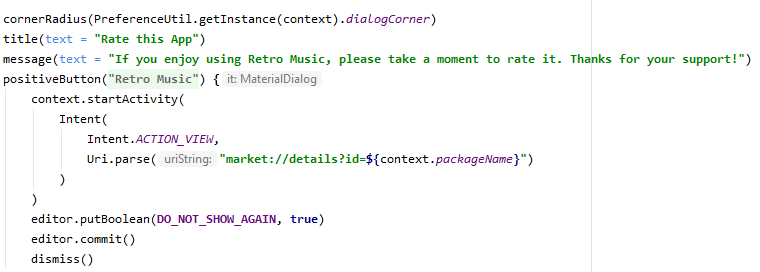
* + - * Al ser la carpeta más funcional, es donde más se vislumbran aportes de terceros, esto por lo general cumplen con buenas prácticas y micro-optimizaciones a lo largo de sus desarrollos, incorporando incluso mejor documentación, que la original.



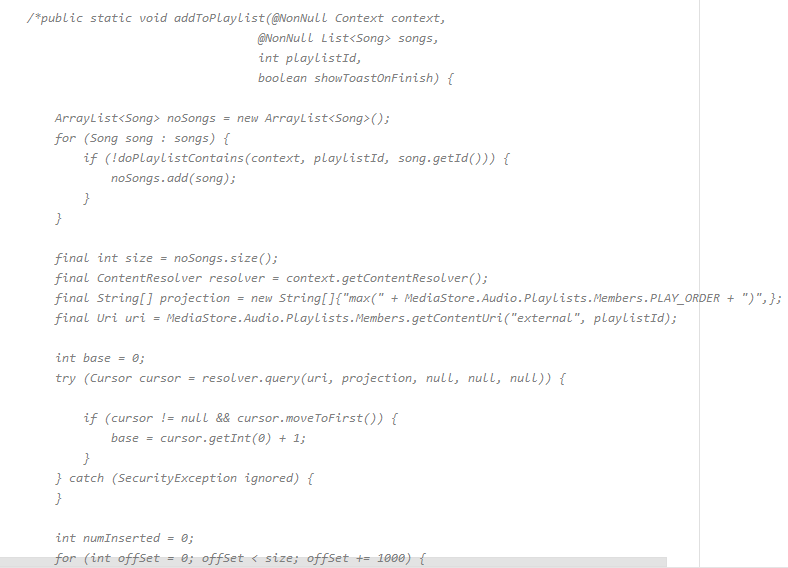
* ThemeManager: Encargado del manejo de themes básicos para la aplicación.
  + - * Vemos una configuración no presente en la visualización del menú “BLACK”.



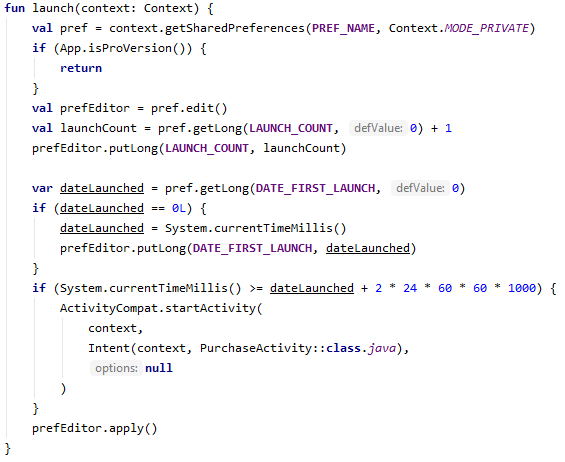
* AppRater: Maneja el dialogo que pide al usuario calificar la aplicación en el PlayStore.
  + - * No tiene manejo de conectividad eventual, por lo que emerge un problema, en caso de no tener acceso a internet, si el usuario indica querer calificar la aplicación, esta intentará redirigir al mismo hacia la correspondiente funcionalidad, sin embargo, al mismo tiempo en Shared Preferences anotará la decisión del usuario, por lo que no volverá a aparecer. Aun si por conectividad eventual no se haya podido finiquitar la calificación.



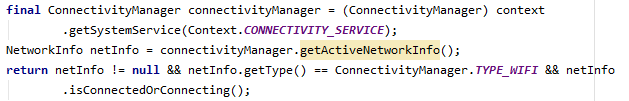
* PlaylistUtil: Encargado del manejo de listas de reproducción.
  + - * Tiene un grupo de aproximadamente 200 líneas de código valido comentadas al final, difícilmente una buena práctica de desarrollo.



* PreferenceUtil: Se encarga de todo el manejo interno de Shared Preferences, así como de la validación de datos en la misma.
* PremiumShow: Se encarga del manejo de verificaciones de la versión pro.



* RetroUtil: Contiene funciones misceláneas de la aplicación, muy poco encapsulada.
  + - * Posee los métodos requeridos para verificar conectividad eventual y evitar bloqueos antes mencionados.



Views:

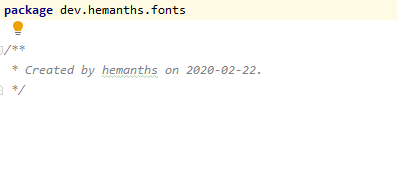
* BreadCrumbLayout: Encargada de incorporar elementos de Bread Crumb Vizualisation.

Volume: Encapsula el manejo y monitoreo de volumen en la aplicación.

App: Maneja y verifica directamente la inicialización de la aplicación.

AppThemeHelper: Encargada de manejar funciones y constantes en referencia a los diferentes themes de la configuración.

Fonts: Código Muerto.



1. Practicas:

A lo largo de la aplicación, hemos notado ciertas prácticas recurrentes. EN el anterior literal, Se llevó a cabo un análisis a escala atómica, clase por clase. Ahora bien, este análisis se llevar a cabo a escala general de la aplicación, apuntando no a las instancias de eventos, sino a patrones de los mismos que se han observado a lo largo de la aplicación.

Podemos ver un espectacular nivel de encapsulamiento. Cada carpeta y clase en su mayoría, tienen una función auto contenida y especifica. Incluso los métodos al interior de las clases cumplen con este patrón. Esta es una excelente práctica. Sin embargo, hemos notado que, a lo largo del desarrollo, se ha experimentado un incremento en el nivel de entropía en el mismo. Vemos unas cuantas apariciones de carpetas y clases inconexas o redundantes. De forma inesperada, los aportes de terceros no eran las fuentes principales de estos errores, sino aportes generados de primera mano por los desarrolladores.

A este respecto, la absoluta falta de documentación, en especial en un proyecto de código abierto es una práctica nefasta para el desarrollo a futuro, así como la mantenibilidad a corto plazo. No era inusual encontrar clases enteras de longitud considerable, completamente habitadas por código funcional y ningún comentario aclarando la funcionalidad de cientos de líneas de código. A la hora de desarrollas sobre o dentro de este tipo de desarrollos, esta práctica puede resultar en costos elevados, e injustificados, completamente eludibles. Podría incluso catalogarse como negligencia, a el nivel presente actualmente.

Podría también mencionar la constante presencia de código muerto, llegando incluso a tener dentro de sí clases muertas. Este tipo de error permite una acumulación de este tipo de entidades, por lo que también es una muy mala práctica a llevar a cabo. Una vez más los aportes de terceros no son la fuente de estos eventos, ya que muchas de estas clases contenían comentarios indicando el autor de las mismas, en todos los casos un miembro del equipo de desarrollo. No es viable mencionar el continuo desarrollo de estas clases como justificación, dado que muchas de ellas fueron creadas hace ya un tiempo considerable.

Adicionalmente, la utilización de tanto Java como Kotlin, en el mismo ambiente de desarrollo, también es considerada una mala práctica. Sin embargo, a comparación de las malas prácticas antes mencionadas, se podría argumentar que el impacto adquirido por esta práctica particular, no es sustancial. Específicamente hablando de la similitud existente entre Java y Kotlin. Aun así, este tipo de prácticas son casi omni-presentes en el código de desarrollo, por lo que mencionarlas no está de más.

1. Conectividad Eventual:

Esta aplicación, tiene un número sustancial de funcionalidades que requieren de conexión a internet. Sin embargo, entre estas, solo hay un grupo muy reducido de funcionalidades claves. Para cada uno de estos casos la aplicación tiene el mismo comportamiento.

La aplicación posee un método en la clase RetroUtils, dedicado a identificar errores de conectividad, sin embargo, en ninguna ocasión la aplicación cambia su comportamiento frente a estos incidentes, fuera de evitar bloquease. Nunca le informa al usuario de este tipo de escenarios, lo cual, en funcionalidades como recuperación de compra, o calificación de la aplicación, conllevan a situaciones malinterpretadas por el usuario.

Está muy lejos de ser una situación ideal, por supuesto. Especialmente dado que en la misma clase RetroUtils, se obtiene un booleano para determinar si hay o no conexión. A partir de este método no debe ser difícil poder responder ante estas situaciones, a partir de sencillos condicionales. Esto mejoraría sustancialmente la experiencia del usuario en referencia a estas funcionalidades.

Se debe anotar, sin embargo, que, en términos de utilización y porcentaje de ejecución, estas funcionalidades ocupan un espacio casi despreciable en el desempeño total de la aplicación. Razón por la cual, se puede deducir, no se han efectuado dichos cambios.