

DESARROLLO DE APLICACIONES PARA DISPOSITIVOS MÓVILES

Metrónomo programable

Daniel Cabal Fernández

Índice de contenido

1	Presentación.....	4
1.1	Descripción del proyecto.....	4
1.2	Objetivos.....	4
1.3	Requerimientos funcionales.....	4
1.3.1	Metrónomo básico.....	4
1.3.2	Programación.....	5
	Cambio de compás.....	5
	Cambio de tempo.....	5
1.3.3	Setlist.....	5
1.4	Posibles ampliaciones.....	5
	Tap tempo.....	5
	Ayudas vocales.....	5
	Exportación a audio.....	6
	Cronómetro.....	6
	Compartición y sincronización entre varios dispositivos.....	6
	Personalización de sonidos.....	6
	Skins.....	6
1.5	Otras cuestiones.....	6
1.5.1	Permisos.....	6
1.5.2	Internacionalización.....	6
1.5.3	Diversidad de dispositivos y rendimiento.....	7
1.6	Requerimientos técnicos.....	7
1.7	Planificación temporal.....	7
1.7.1	Diagrama de Gantt.....	8
1.8	Observaciones adicionales.....	8
2	Análisis.....	9
2.1	Descripción de usuarios.....	9
2.1.1	Usuario tipo.....	9
2.1.2	Usuarios para las pruebas.....	9
2.2	Requerimientos funcionales.....	10
2.2.1	Metrónomo básico.....	10
	Caso de uso selección de tempo.....	10
	Caso de uso selección de compás.....	11
	Caso de uso modificación de click.....	11
	Caso de uso reproducción.....	11
	Caso de uso detener reproducción.....	12
2.2.2	Metrónomo programable.....	13
	Caso de uso crear canción.....	16
	Caso de uso reproducir canción.....	17
	Caso de uso eliminar canción.....	17
	Caso de uso editar canción.....	17
2.2.3	Listas de reproducción.....	19
	Caso de uso crear lista de reproducción.....	19
	Caso de uso ordenar lista de reproducción.....	20
	Caso de uso eliminar lista de reproducción.....	21

Caso de uso eliminar canción de lista de reproducción.....	21
Caso de uso añadir canción a lista de reproducción.....	21
2.3 Interfaz de usuario.....	23
2.3.1 Pantalla principal.....	23
2.3.2 Canciones.....	29
2.3.3 Listas de reproducción.....	32
2.3.4 Secuencia de pantallas.....	35
2.4 Diseño de arquitectura.....	38
2.4.1 Diagrama detallado.....	39
2.5 Persistencia.....	42
2.6 Pruebas previas.....	44
2.6.1 Descripción.....	44
2.6.2 Resultados.....	44
3 Implementación.....	45
3.1 Estructura.....	45
MetroProg.js.....	45
MPManager.js.....	45
Sequence.js.....	45
Song.js.....	45
SongManager.js.....	46
Setlist.js.....	46
SetlistManager.js.....	46
3.2 Problemas de rendimiento.....	46
3.3 Aspectos pendientes de finalización.....	49
Splash screen.....	49
Integración de diálogos.....	49
Ordenación de listas de reproducción.....	49
Orientación de la pantalla.....	49
Bloqueo de pantalla.....	49
4 Conclusiones.....	49
ANEXO.....	51
Usuario 1.....	51
Usuario 2.....	52
Usuario 3.....	53

1 Presentación

1.1 Descripción del proyecto

El metrónomo es una herramienta fundamental en el día a día de cualquier músico, desde el aprendizaje más básico de principiantes hasta estudios avanzados de instrumentistas experimentados. Además se hace imprescindible su uso en grabaciones y cada vez con más frecuencia en las actuaciones en directo.

Como músico en activo, he utilizado diferentes recursos para mantener el tempo en horas de estudio, en grabaciones y en actuaciones en vivo; desde pequeños metrónomos digitales, programaciones en secuenciadores por software, y más recientemente metrónomos internos en baterías electrónicas y aplicaciones para teléfonos móviles.

La función de esta herramienta es marcar el tempo de una forma fiable y constante, y opcionalmente, marcar con un sonido diferente los comienzos de compás o las subdivisiones de éste.

El problema que me he encontrado muchas veces es la carencia de algún tipo de función de programación. Generalmente, en la mayoría de composiciones de pop, rock o metal no suele ser habitual encontrarse con cambios de tempo o de compás, pero sí es muy habitual en otras más complejas de estilos como el rock progresivo, donde podemos encontrarnos con temas muy largos donde lo habitual son multitud de cambios de compás, compases irregulares o de amalgama y cambios de tempo. En este caso, tanto en sesiones de grabación como en actuaciones en directo, se hace fundamental el uso de alguna herramienta que permita programar con exactitud todas estas variaciones. De otra forma, se puede hacer muy tedioso el trabajo de realizarlo en algún tipo de secuenciador por software, para posteriormente exportarlo a algún formato habitual de audio y reproducirlo desde algún dispositivo portátil (ordenador, tablet, móvil, reproductor de música portátil, etc).

1.2 Objetivos

- Iniciarse en el campo del desarrollo de aplicaciones para dispositivos móviles.
- Adaptar los conocimientos personales de HTML, CSS y Javascript al desarrollo de aplicaciones móviles.
- Poner en práctica los métodos aprendidos en la asignatura Técnicas de Desarrollo de Software, entre otras.
- Desarrollar una aplicación totalmente funcional con el objetivo de publicarla en las distintas plataformas de aplicaciones móviles.

1.3 Requerimientos funcionales

1.3.1 Metrónomo básico

La primera funcionalidad de la aplicación será un metrónomo básico, en el que se indicará

un tempo y un compás y simplemente se pulsará la reproducción que se mantendrá hasta que se pause manualmente.

El metrónomo será capaz de reproducir cualquier compás de blancas, negras, corcheas y semicorcheas (del tipo $x/2$, $x/4$, $x/8$ y $x/16$) marcando por defecto la primera parte fuerte y el resto débiles. El usuario podrá ser capaz de cambiar éste comportamiento fácilmente, marcando cualquier parte como fuerte, débil o anulando su sonido.

1.3.2 Programación

El usuario podrá programar una canción completa. Para ello dispondrá de las siguientes posibilidades:

Cambio de compás

Se podrán programar tantos cambios de compás como se desee de forma secuencial, es decir se podrá programar algo del tipo 16 compases de $4/4$ – 8 compases de $7/8$ – 1 compás de $5/4$ – un compás de $7/4$ – un compás de $5/8$.

Cambio de tempo

Al igual que los cambios de compás se podrán introducir cambios de tempo sin restricción en la misma canción: 32 compases a 120 bpm – 16 compases a 95 bpm – 12 compases a 140 bpm.

1.3.3 Setlist

El usuario podrá guardar diferentes canciones programadas dándoles a cada una de ellas un título. Estas canciones podrán ser ordenadas en diferentes *setlists* o listas de canciones para directo que también se distinguirán por su nombre. Se deberán poder reordenar de manera sencilla, y cada canción puede estar en uno, varios o ningún *setlist*.

Un *setlist* funcionará como una lista de reproducción de cualquier reproductor de música, con la salvedad de que una vez finalice una canción, la reproducción se detendrá, y la siguiente se tendrá que reproducir de forma manual.

1.4 Posibles ampliaciones

Tap tempo

La funcionalidad *tap tempo* es muy habitual en metrónomos, osciladores, arpegiadores, etc. Consiste en marcar el tempo de forma manual sobre un botón. El aparato (en este caso la aplicación) calcula el tempo de forma automática en base a las pulsaciones del usuario.

Ayudas vocales

Utilizando alguna aplicación externa de grabación de voz, se podrían sincronizar instrucciones junto con la canción. Estas instrucciones serían del tipo “guitarra entra en 4,

3, 2, 1" y su utilización es bastante común.

Exportación a audio

En grabaciones de estudio es muy común utilizar en los sistemas multipistas una pista de audio con la claqueta. La aplicación sería capaz de exportar las canciones guardadas en uno o varios de los formatos de audio más comunes para posteriormente importarlo en cualquiera de estos sistemas, o bien reproducirlos en otro dispositivo como cualquier otro archivo de audio.

Cronómetro

Mostrar mediante un cronómetro la duración de la programación realizada. También se podría utilizar de forma inversa, es decir, introduciendo un tiempo y la aplicación generaría el número de compases exacto al tempo especificado.

Compartición y sincronización entre varios dispositivos

Una funcionalidad interesante sería poder compartir las canciones y listas de reproducción entre dispositivos con la finalidad de poder iniciar y parar la reproducción sincronizada desde uno sólo de ellos. La dificultad que podemos encontrar es que un metrónomo es una herramienta que debe ser muy precisa, y por tanto la reproducción se debe iniciar a la vez en todos los dispositivos, pues de lo contrario perdería totalmente su función.

Personalización de sonidos

Simplemente se pueden añadir más variedad de sonidos para el "click", permitiendo al escoger entre ellos.

Skins

Crear diferentes skins que se puedan seleccionar en un apartado de preferencias.

1.5 Otras cuestiones

1.5.1 Permisos

Las aplicaciones para dispositivos móviles requieren ciertos permisos que el usuario ha de revisar y aceptar para poder instalar y utilizarlas. En el caso de la aplicación que planteo no se requerirá de ningún permiso especial; en principio, solamente necesitaría permisos de lectura y escritura sobre los medios de almacenamiento del dispositivo y bloqueo de pantalla para que no entre en reposo durante la reproducción.

1.5.2 Internacionalización

Debe ser necesario que la aplicación esté disponible en la mayor cantidad de idiomas posible, dado que todas las tiendas de las distintas plataformas son internacionales y

distribuyen aplicaciones en todo el mundo.

1.5.3 Diversidad de dispositivos y rendimiento

La gran variedad de dispositivos móviles en el mercado con características técnicas tan dispares obliga a que la aplicación sea ligera para poder funcionar con fluidez en tantos como sea posible. Sin un funcionamiento fluido, el metrónomo se hace totalmente inútil.

Las pruebas se tratarán de realizar en la mayor cantidad posible de dispositivos, pero en primera instancia se realizarán sobre dispositivos Android.

1.6 Requerimientos técnicos

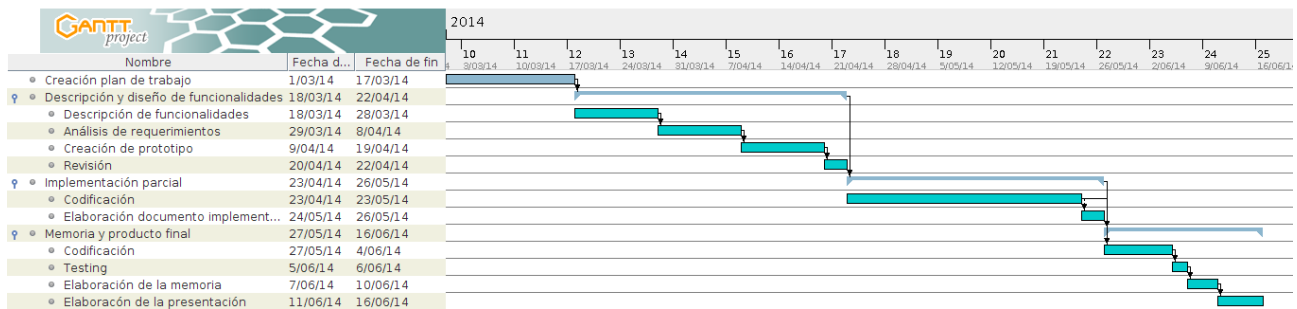
- NetBeans IDE para la codificación
- Cordova y Android SDK para la compilación
- Adobe Photoshop para el prototipado
- Dia para los diferentes diagramas a realizar
- GanttProject para la elaboración de diagramas Gantt
- LibreOffice para la redacción de documentación

1.7 Planificación temporal

Tarea	Inicio	Fin
Fase 1. Análisis previo	01/03/2014	01/03/2014
Creación plan de trabajo	01/03/2014	17/03/2014
Entrega PEC 1	17/03/2014	17/03/2014
Fase 2. Descripción y diseño de funcionalidades	18/03/2014	22/04/2014
Descripción de funcionalidades	18/03/2014	28/03/2014
Análisis de requerimientos	29/03/2014	08/04/2014
Creación de prototipo	09/04/2014	19/04/2014
Revisión	20/04/2014	22/04/2014
Entrega PEC 2	22/04/2014	22/04/2014
Fase 3. Implementación parcial	23/04/2014	26/05/2014
Codificación	23/04/2014	23/05/2014
Elaboración documento implementación	24/05/2014	26/05/2014
Entrega PEC 3	26/05/2014	26/05/2014
Memoria y producto final	27/05/2014	16/06/2014
Terminar la codificación (producto final)	27/05/2014	04/06/2014
Testing	05/06/2014	06/06/2014
Elaboración de la memoria	07/06/2014	10/06/2014
Elaboración de la presentación	11/06/2014	16/06/2014
Entrega memoria	16/06/2014	16/06/2014

Las fechas de esta planificación son aproximadas, y se podrán ajustar a las necesidades de cada parte del proyecto.

1.7.1 Diagrama de Gantt



1.8 Observaciones adicionales

Se han descrito un importante número de funcionalidades adicionales o posibles ampliaciones. El objetivo de este proyecto es desarrollar las funcionalidades básicas y el mayor número posible de ampliaciones, sin embargo, éstas estarán siempre supeditadas al tiempo que implique desarrollar las básicas, pudiendo añadirse alguna funcionalidad descrita en las ampliaciones o cancelarse algunas ya definidas a lo largo de la duración del proyecto.

2 Análisis

2.1 Descripción de usuarios

2.1.1 Usuario tipo

El usuario tipo de la aplicación es el siguiente:

- Músico aficionado, estudiante o profesional, con unos conocimientos dispares de teoría de la música -concretamente para utilizar la aplicación basta con conocer los términos *tempo*¹, *pulso*² y *compás*³.
- Frecuencia de utilización muy variable, desde usuarios que lo utilicen ocasionalmente hasta otros que lo puedan utilizar varias horas al día.
- De edad también variable, estaría dentro de la edad de cualquier usuario de smartphone.
- El hardware que utilizarán serán principalmente smartphones, aunque ocasionalmente puedan utilizar tablets.

2.1.2 Usuarios para las pruebas

Los usuarios elegidos para las pruebas son tres de los componentes del grupo de Nu Folk asturiano Xera⁴, con amplia experiencia musical y conocimientos:

- Elena Pinto. Teclista, licenciada en Historia y Ciencias de la Música. Profesora de Educación Secundaria Obligatoria. 40 años.
- Flavia Rodríguez. Violinista, licenciada en Violín y en Historia y Ciencias de la Música. Máster en interpretación de violín. 30 años.
- Carlos Díaz. Cantante, bouzoukista y pianista. Técnico de sonido. Diplomado en Magisterio especialidad Educación Musical. 34 años.

2.2 Requerimientos funcionales

A continuación se describen los requerimientos y casos de uso de cada una de las funcionalidades básicas, así como de las posibles ampliaciones.

2.2.1 Metrónomo básico

La principal funcionalidad de la aplicación es la de un metrónomo básico en la que se fija un tempo, un compás y los *clicks* fuertes y débiles. Por defecto se iniciará a 100 bpm en

1 <http://es.wikipedia.org/wiki/Tempo>

2 http://es.wikipedia.org/wiki/Pulso_%28m%C3%BAsica%29

3 http://es.wikipedia.org/wiki/Comp%C3%A1s_%28m%C3%BAsica%29

4 <http://xera.eu/>

compás de 4/4, siendo el *click* fuerte el 1 y los débiles el 2, 3 y 4. El usuario iniciará y detendrá la reproducción manualmente.

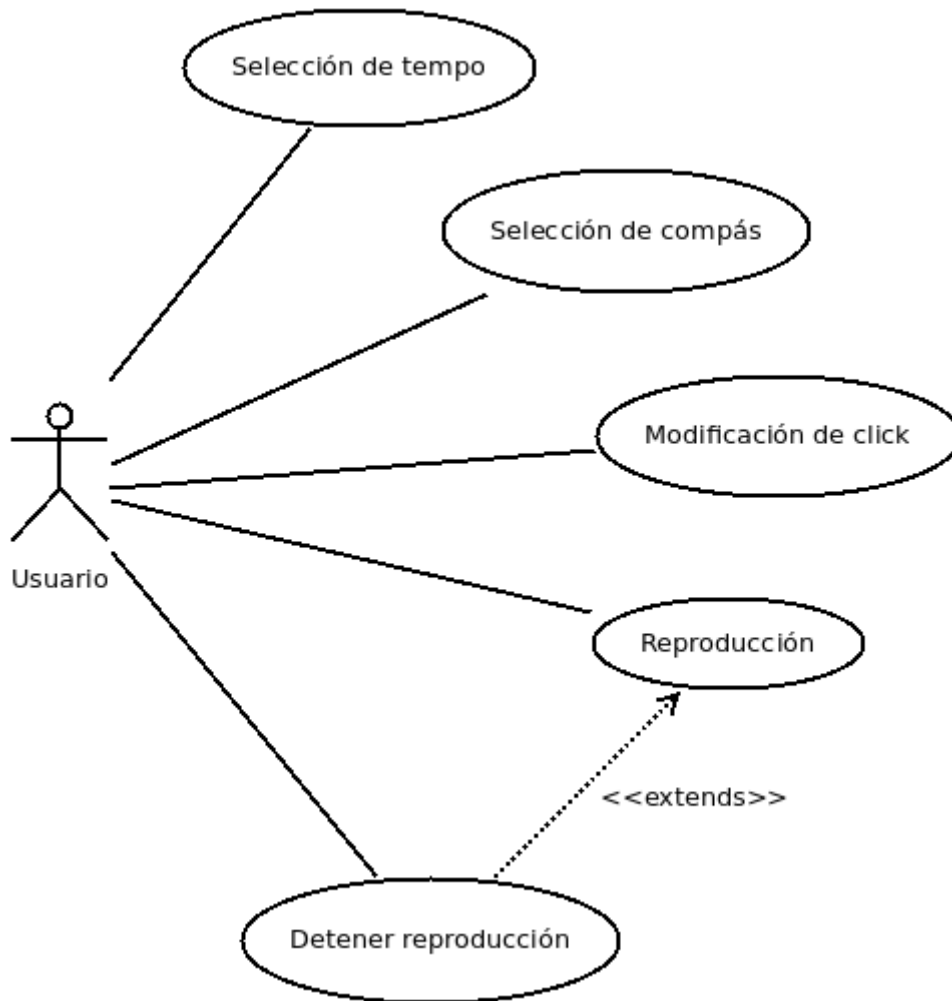


Ilustración 1: Casos de uso metrónomo básico

Caso de uso selección de tiempo

Descripción	Selección de tiempo
Actores	Usuario
Casos de uso relacionados	
Precondición	
Postcondición	Se ha fijado un tiempo diferente al mostrado por defecto
Proceso normal	<ol style="list-style-type: none"> 1. El sistema muestra la pantalla principal 2. El actor pulsa sobre el número que indica el tiempo 3. El sistema muestra una ventana emergente con un teclado numérico y el valor del tiempo 4. El actor introduce el nuevo valor 5. El sistema vuelve a la pantalla principal con el valor introducido por el actor
Alternativas de proceso y excepciones	<ol style="list-style-type: none"> 4a1. El actor cancela la acción 4a2. El sistema vuelve a la pantalla principal sin cambiar el tiempo

Caso de uso selección de compás

Descripción	Selección de compás
Actores	Usuario
Casos de uso relacionados	
Precondición	
Postcondición	Se ha fijado un compás diferente al mostrado por defecto
Proceso normal	<ol style="list-style-type: none"> 1. El sistema muestra la pantalla principal 2. El actor pulsa sobre el botón que indica el compás 3. El sistema muestra una ventana emergente donde se seleccionará de forma independiente el número superior (número de notas) y el número inferior (duración de las notas) 4. El actor selecciona los nuevos valores 5. El sistema vuelve a la pantalla principal con el compás seleccionado por el actor
Alternativas de proceso y excepciones	<p>4a1. El actor cancela la acción</p> <p>4a2. El sistema vuelve a la pantalla principal sin cambiar el compás</p>

Caso de uso modificación de *click*

Descripción	Modificación de los pulsos fuertes y débiles del compás
Actores	Usuario
Casos de uso relacionados	
Precondición	
Postcondición	Se han modificado los pulsos fuertes y débiles por defecto
Proceso normal	<ol style="list-style-type: none"> 1. El sistema muestra la pantalla principal 2. El actor pulsa sobre cualquiera de los pulsos representados en pantalla 3. El sistema modifica el pulso secuencialmente (fuerte-débil-mudo) y lo cambia de color (rojo-verde-gris respectivamente)
Alternativas de proceso y excepciones	

Caso de uso reproducción

Descripción	Se inicia la reproducción
Actores	Usuario
Casos de uso relacionados	Detener reproducción
Precondición	La aplicación está en posición "Stop"
Postcondición	Se ha iniciado la reproducción
Proceso normal	<ol style="list-style-type: none"> 1. El actor pulsa el botón "Play" 2. El sistema inicia la reproducción 3. El sistema pasa a posición "Play" cambiando el icono del botón de inicio/fin
Alternativas de proceso y excepciones	

Caso de uso detener reproducción

Descripción	Se detiene la reproducción
Actores	Usuario
Casos de uso relacionados	Reproducción
Precondición	La aplicación está en posición "Play"
Postcondición	Se ha detenido la reproducción
Proceso normal	<ol style="list-style-type: none"> 1. El actor pulsa el botón "Stop" 2. El sistema detiene la reproducción 3. El sistema pasa a posición "Stop" cambiando el icono del botón de inicio/fin
Alternativas de proceso y excepciones	

2.2.2 Metrónomo programable

A continuación se describen los casos de uso propios de la parte programable del metrónomo. En este caso se crean canciones, indicando de forma secuencial número de compases, tipo de compás y tiempo. La canción se podrá guardar para poder reproducirla o modificarla en cualquier momento.

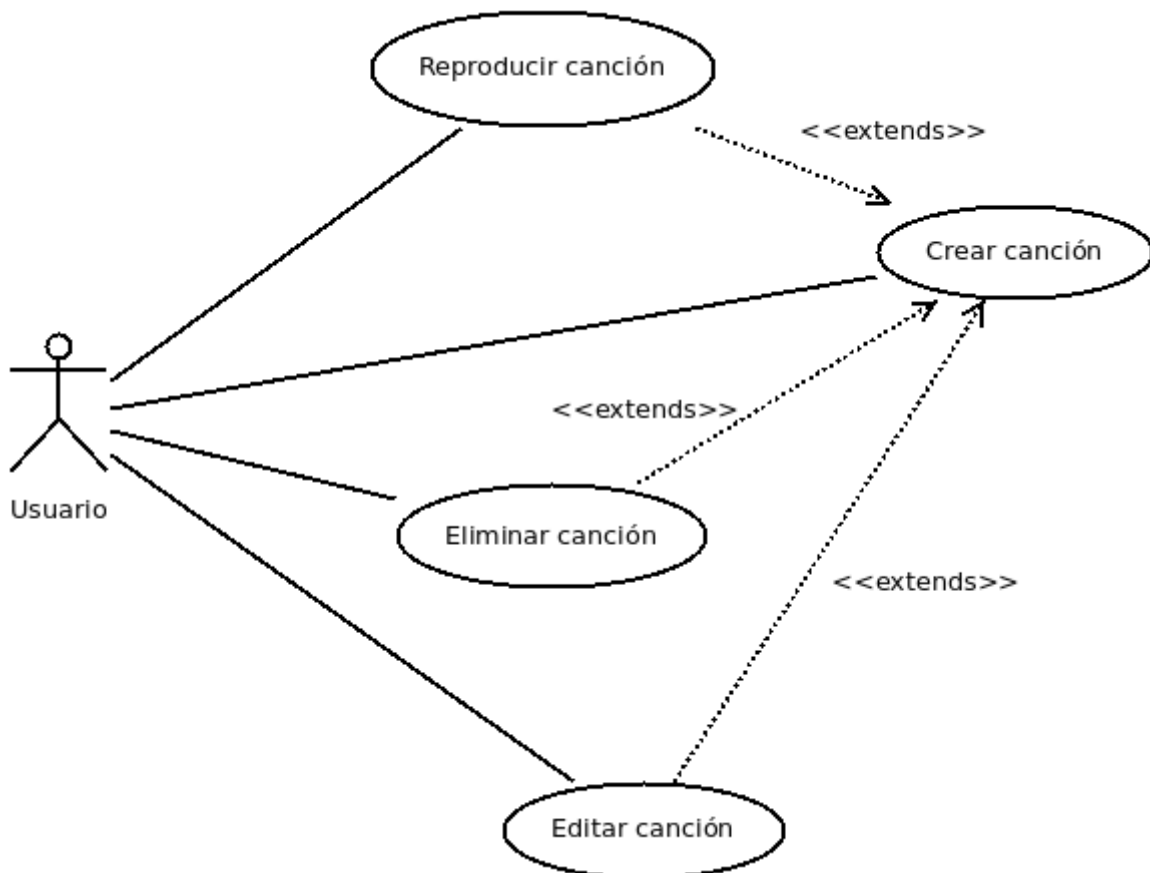


Ilustración 2: Casos de uso metrónomo programable

Caso de uso crear canción

Descripción	Se crea una canción con una secuencia de compases y tempos
Actores	Usuario
Casos de uso relacionados	Reproducir canción Eliminar canción Editar canción
Precondición	
Postcondición	El sistema guarda una canción
Proceso normal	<ol style="list-style-type: none"> 1. El actor pulsa en el menú de opciones → canción 2. El sistema muestra un listado con las canciones guardadas y botones para añadir una canción nueva, reproducir, editar y eliminar una canción existente 3. El actor pulsa sobre el botón de canción nueva 4. El sistema solicita el nombre de la nueva canción 5. El actor introduce el nombre de la canción y pulsa aceptar 6. El sistema muestra una ventana emergente donde pide el número de compases, el tipo de compás, la selección de <i>clicks</i> fuertes y débiles y el tempo de la sección. 7. El actor introduce los datos requeridos 8. El sistema muestra una pantalla con las secuencias introducidas 9. El actor pulsa finalizar 10. El sistema guarda la canción y vuelve a la pantalla principal
Alternativas de proceso y excepciones	<ol style="list-style-type: none"> 3a1. El actor cancela la operación 3a2. El sistema vuelve a la pantalla que muestra el listado de canciones 5a1. El actor cancela la operación 5a2. El sistema vuelve a la pantalla que muestra el listado de canciones 7a1. El actor cancela la operación 7a2. El sistema vuelve a la pantalla que muestra el listado de canciones

Caso de uso reproducir canción

Descripción	Se reproduce una canción
Actores	Usuario
Casos de uso relacionados	Crear canción Eliminar canción Editar canción
Precondición	Se debe haber guardado alguna canción
Postcondición	Se ha reproducido una canción
Proceso normal	<ol style="list-style-type: none"> 1. El actor pulsa en el menú de opciones → canción 2. El sistema muestra un listado con las canciones guardadas y botones para añadir una canción nueva, reproducir, editar y eliminar una canción existente 3. El actor selecciona una canción y pulsa el botón de reproducción 4. El sistema carga la canción seleccionada y se mantiene a la espera 5. El actor pulsa el botón de reproducción 6. El sistema reproduce la canción
Alternativas de proceso y excepciones	<ol style="list-style-type: none"> 3a1. El actor cancela la operación 3a2. El sistema vuelve a la pantalla principal

Caso de uso eliminar canción

Descripción	Se elimina una canción
Actores	Usuario

Casos de uso relacionados	Crear canción Reproducir canción Editar canción
Precondición	Se debe haber guardado alguna canción
Postcondición	Se ha eliminado una canción
Proceso normal	<ol style="list-style-type: none"> 1. El actor pulsa en el menú de opciones → canción 2. El sistema muestra un listado con las canciones guardadas y botones para añadir una canción nueva, reproducir, editar y eliminar una canción existente 3. El actor selecciona una canción y pulsa el botón de borrar 4. El sistema muestra una ventana de confirmación 5. El actor pulsa el botón de confirmación 6. El sistema elimina la canción
Alternativas de proceso y excepciones	<ol style="list-style-type: none"> 3a1. El actor cancela la operación 3a2. El sistema vuelve a la pantalla principal 5a1. El actor cancela la operación 5a2. El sistema vuelve a la pantalla donde se muestra el listado de canciones

Caso de uso editar canción

Descripción	Se modifican los datos de una canción
Actores	Usuario
Casos de uso relacionados	Crear canción Reproducir canción Eliminar canción
Precondición	Se debe haber guardado alguna canción
Postcondición	Se ha eliminado una canción
Proceso normal	<ol style="list-style-type: none"> 1. El actor pulsa en el menú de opciones → canción 2. El sistema muestra un listado con las canciones guardadas y botones para añadir una canción nueva, reproducir, editar y eliminar una canción existente 3. El actor selecciona una canción y pulsa el botón de editar 4. El sistema muestra una ventana con la secuencia de parámetros de la canción 5. El actor pulsa sobre uno de ellos 6. El sistema muestra la ventana de edición con el tipo de compás número de compases y tempo 7. El actor modifica los parámetros y acepta los cambios 8. El sistema muestra la ventana con la secuencia de parámetros de la canción 9. El actor acepta los cambios 10. El sistema vuelve a la pantalla donde se muestra el listado de canciones.
Alternativas de proceso y excepciones	<ol style="list-style-type: none"> 3a1. El actor cancela la operación 3a2. El sistema vuelve a la pantalla principal 5a1. El actor cancela la operación 5a2. El sistema vuelve a la pantalla donde se muestra el listado de canciones 7a1. El actor cancela la operación 7a2. El sistema vuelve a la pantalla con la secuencia de parámetros de la canción

2.2.3 Listas de reproducción

Las listas de reproducción son especialmente útiles en actuaciones en vivo, y su funcionamiento es muy similar al de cualquier reproductor multimedia. En este apartado se describen los casos de uso relacionados con esta funcionalidad; la creación, edición y borrado de listas de reproducción.

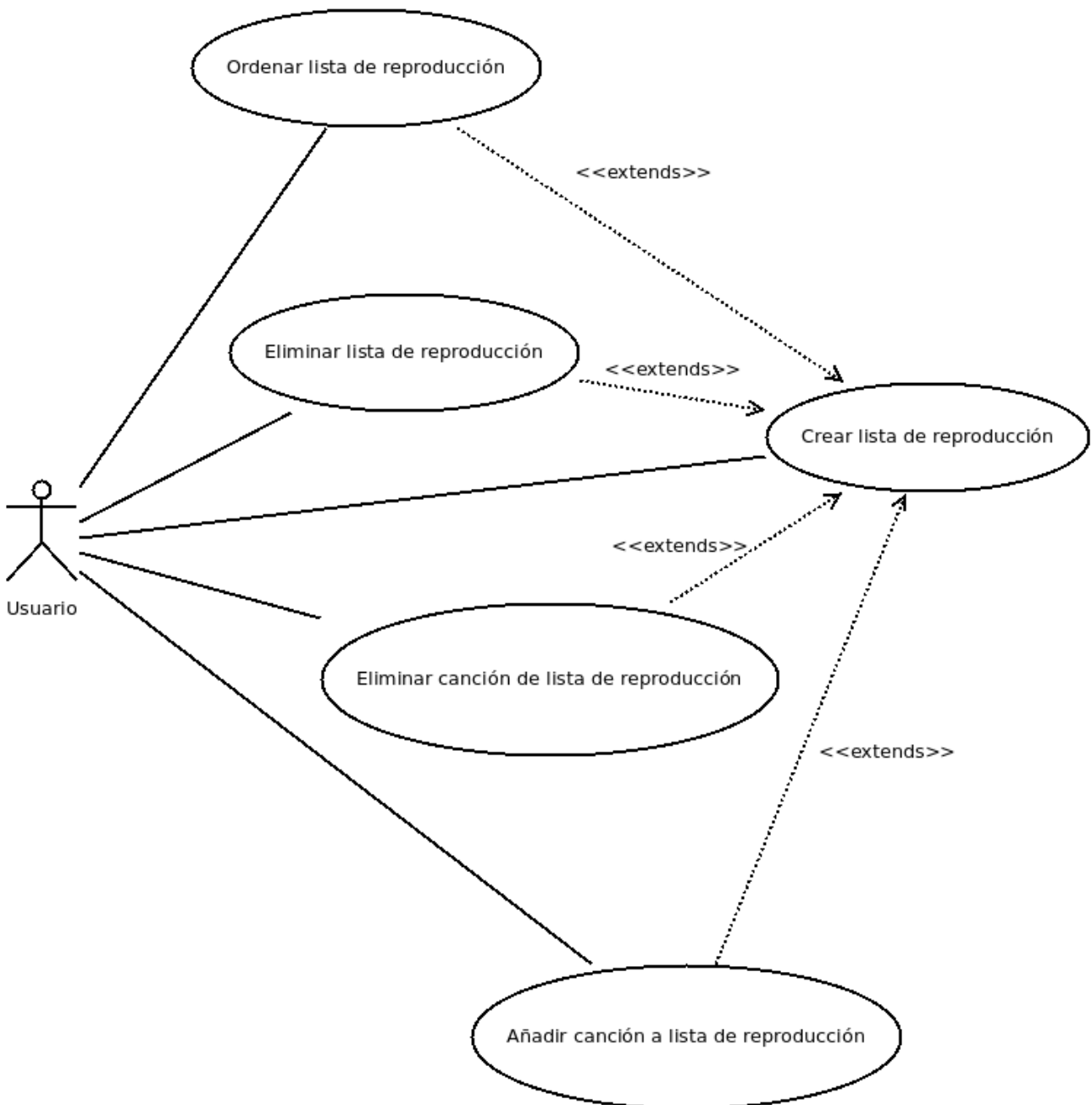


Ilustración 3: Casos de uso listas de reproducción

Caso de uso crear lista de reproducción

Descripción	Se crea una nueva lista de reproducción
Actores	Usuario
Casos de uso relacionados	Ordenar lista de reproducción Eliminar lista de reproducción Eliminar canción de lista de reproducción Añadir canción a lista de reproducción
Precondición	Debe haber al menos una canción guardada

Postcondición	Se ha creado una lista de reproducción
Proceso normal	<ol style="list-style-type: none"> 1. El actor pulsa en el menú de opciones → setlist 2. El sistema muestra una relación de listas de reproducción y una serie de botones para añadir, eliminar o editar listas de reproducción 3. El actor pulsa en el botón de añadir lista de reproducción 4. El sistema muestra una ventana donde solicita el nombre de la lista de reproducción 5. Una vez introducido el nombre, el sistema muestra un listado con las canciones guardadas 6. El actor selecciona las canciones que desea añadir a la lista 7. El actor finaliza la selección 8. El sistema muestra el listado de canciones añadidas a la lista de reproducción 9. El actor ordena la lista a su gusto pulsando sobre una canción y arrastrándola hacia la posición deseada 10. El actor confirma los cambios 11. El sistema vuelve a la pantalla principal
Alternativas de proceso y excepciones	<ol style="list-style-type: none"> 5a1. El actor cancela la operación 5a2. El sistema vuelve al listado de canciones guardadas 6a1. El actor cancela la operación 6a2. El sistema vuelve al listado de canciones guardadas 7a1. El actor cancela la operación 7a2. El sistema vuelve al listado de canciones guardadas 9a1. El actor confirma la lista de reproducción sin ordenar la lista 9a2. El sistema vuelve a la pantalla principal

Caso de uso ordenar lista de reproducción

Descripción	Se cambia el orden de las canciones en una lista de reproducción
Actores	Usuario
Casos de uso relacionados	<p>Crear lista de reproducción Eliminar lista de reproducción Eliminar canción de lista de reproducción Añadir canción a lista de reproducción</p>
Precondición	Debe existir al menos una lista de reproducción
Postcondición	Se ha cambiado el orden de canciones de una lista de reproducción
Proceso normal	<ol style="list-style-type: none"> 1. El actor pulsa en el menú de opciones → setlist 2. El sistema muestra una relación de listas de reproducción y una serie de botones para añadir, eliminar o editar listas de reproducción 3. El actor ordena la lista a su gusto pulsando sobre una canción y arrastrándola hacia la posición deseada 4. El actor confirma los cambios 5. El sistema vuelve a la pantalla principal
Alternativas de proceso y excepciones	<ol style="list-style-type: none"> 2a1. El actor cancela la operación 2a2. El sistema vuelve a la pantalla principal 4a1. El actor cancela la operación 4a2. El sistema vuelve a la pantalla principal

Caso de uso eliminar lista de reproducción

Descripción	Se elimina una de las listas de reproducción guardadas
--------------------	--

Actores	Usuario
Casos de uso relacionados	Crear lista de reproducción Ordenar lista de reproducción Eliminar canción de lista de reproducción Añadir canción a lista de reproducción
Precondición	Debe existir al menos una lista de reproducción
Postcondición	Se ha eliminado una lista de reproducción
Proceso normal	<ol style="list-style-type: none"> 1. El actor pulsa en el menú de opciones → setlist 2. El sistema muestra una relación de listas de reproducción y una serie de botones para añadir, eliminar o editar listas de reproducción 3. El actor selecciona una lista de reproducción y pulsa el botón eliminar 4. El sistema muestra una ventana de confirmación 5. El actor confirma la acción 6. El sistema elimina la lista de reproducción y vuelve al catálogo de listas de reproducción
Alternativas de proceso y excepciones	<p>5a1. El actor cancela la operación</p> <p>5a2. El sistema vuelve a la pantalla con la relación de listas de reproducción</p>

Caso de uso eliminar canción de lista de reproducción

Descripción	Se elimina una o varias canciones de las contenidas en una lista de reproducción
Actores	Usuario
Casos de uso relacionados	Crear lista de reproducción Ordenar lista de reproducción Eliminar lista de reproducción Añadir canción a lista de reproducción
Precondición	Debe existir al menos una lista de reproducción con al menos una canción
Postcondición	Se ha eliminado una canción de la lista de reproducción
Proceso normal	<ol style="list-style-type: none"> 1. El actor pulsa en el menú de opciones → setlist 2. El sistema muestra una relación de listas de reproducción y una serie de botones para añadir, eliminar o editar listas de reproducción 3. El actor selecciona una lista de reproducción y pulsa el botón editar 4. El sistema muestra la lista de canciones y unos botones para añadir o eliminar canciones a la lista 5. El actor selecciona una de las canciones y pulsa eliminar 6. El sistema muestra una pantalla con un mensaje de confirmación 7. El actor confirma la acción 8. El sistema elimina la canción de la lista de reproducción y muestra la lista modificada
Alternativas de proceso y excepciones	<p>7a1. El actor cancela la acción</p> <p>7a2. El sistema muestra la relación de canciones de la lista de reproducción</p>

Caso de uso añadir canción a lista de reproducción

Descripción	Se elimina una o varias canciones de las contenidas en una lista de reproducción
Actores	Usuario
Casos de uso relacionados	Crear lista de reproducción Ordenar lista de reproducción Eliminar lista de reproducción

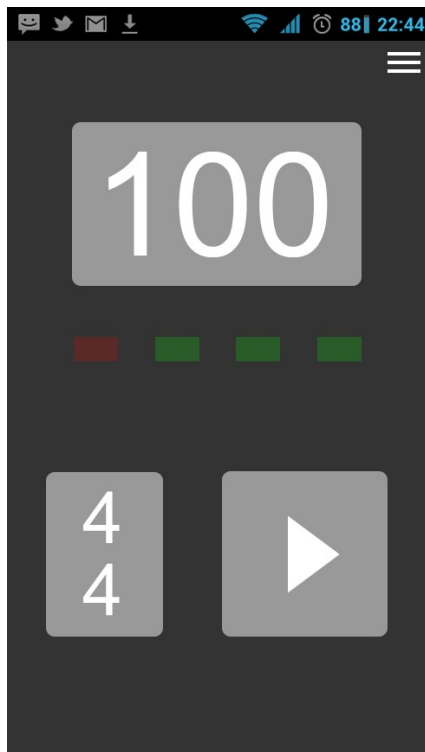
	Eliminar canción de lista de reproducción
Precondición	Debe existir al menos una lista de reproducción
Postcondición	Se ha añadido una canción de la lista de reproducción
Proceso normal	<ol style="list-style-type: none"> 1. El actor pulsa en el menú de opciones → setlist 2. El sistema muestra una relación de listas de reproducción y una serie de botones para añadir, eliminar o editar listas de reproducción 3. El actor selecciona una lista de reproducción y pulsa el botón editar 4. El sistema muestra la lista de canciones y unos botones para añadir o eliminar canciones a la lista 5. El actor pulsa el botón añadir 6. El sistema muestra el listado de canciones guardadas 7. El actor selecciona las canciones deseadas y pulsa el botón finalizar 8. El sistema añade las canciones seleccionadas a la lista de reproducción y
Alternativas de proceso y excepciones	<ol style="list-style-type: none"> 7a1. El actor cancela la acción 7a2. El sistema muestra la relación de canciones de la lista de reproducción

2.3 Interfaz de usuario⁵

2.3.1 Pantalla principal

La pantalla principal muestra la información sobre tempo y compás. También aparecen las pulsaciones del compás que se iluminarán durante la reproducción: Rojo para *click* fuerte, verde para *click* débil y gris sin *click*. Durante la reproducción de una canción, aparece también el nombre de ésta.

⁵ Las pantallas mostradas en esta sección tienen el *look and feel* de la versión 4.2 de Android, no obstante, la aplicación se desarrollará utilizando el framework jQuery Mobile, que tiene un aspecto diferente, pero se respetará la distribución de elementos en todas las pantallas.

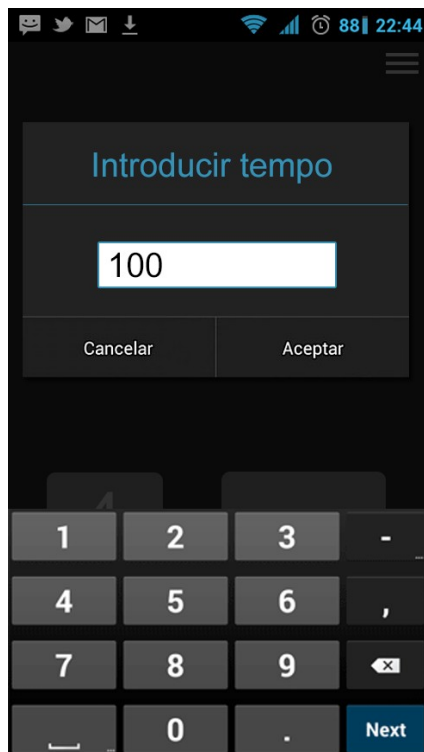


4. Ilustración: Pantalla principal

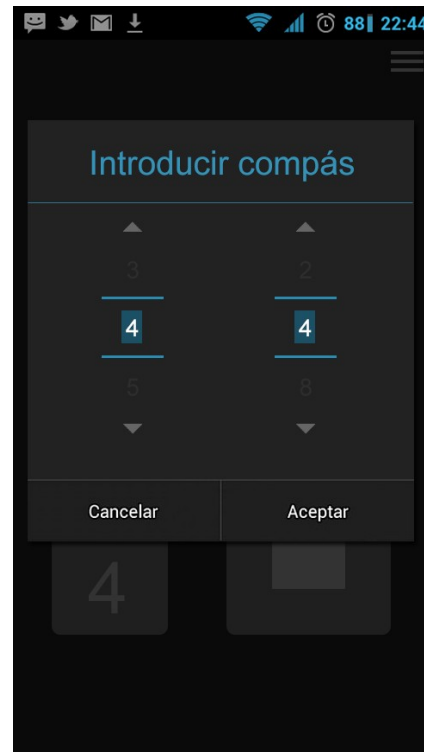


5. Ilustración: Pantalla principal durante la reproducción de una canción

Desde esta pantalla se puede acceder de forma directa a la selección de tiempo y compás pulsando sobre sus respectivos indicadores. La modificación del *click* se hace directamente pulsando sobre las “luces” de los tiempos. Estas acciones no se podrán realizar durante la reproducción de una canción.



6. Ilustración: Selección de tempo



7. Ilustración: Selección de compás

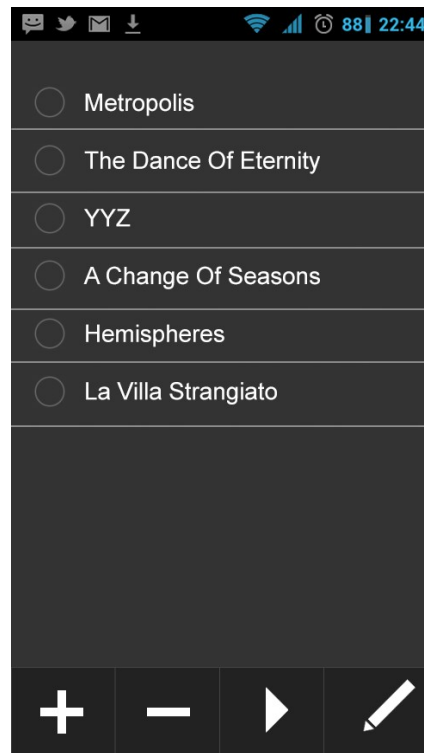
Pulsando sobre el icono que aparece en la parte superior derecha se accede a las secciones de canción, *setlist* (lista de reproducción) y se vuelve a la pantalla inicial desde cualquiera de las otras.



8. Ilustración: Pantalla principal con el menú de opciones desplegado

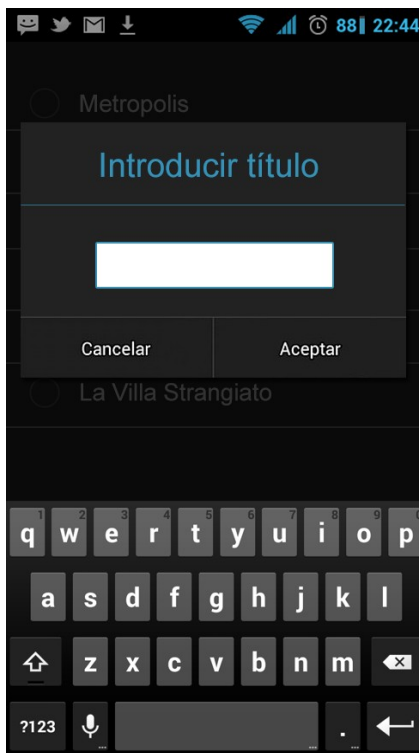
2.3.2 Canciones

Una vez pulsada la opción *canción*, aparecerá la siguiente pantalla. Mediante los botones de la parte inferior se podrá crear una nueva canción o eliminar, editar o reproducir una de las canciones guardadas una vez seleccionada.

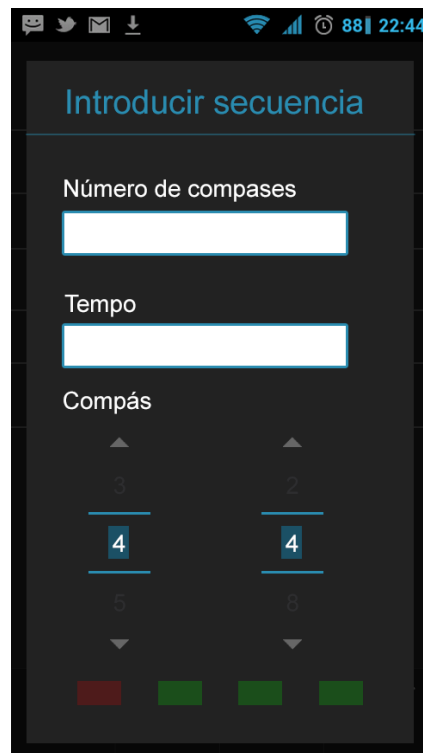


9. Ilustración: Pantalla "canción"

Para crear una canción primeramente debemos introducir su título. Una vez hecho esto, la canción se creará de forma secuencial, es decir, se introducirán ordenadamente los diferentes cambios de tempo y/o compás además del número de compases en los que se mantiene. También se podrán modificar los golpes fuertes y débiles del compás que se indican por defecto para cada secuencia.

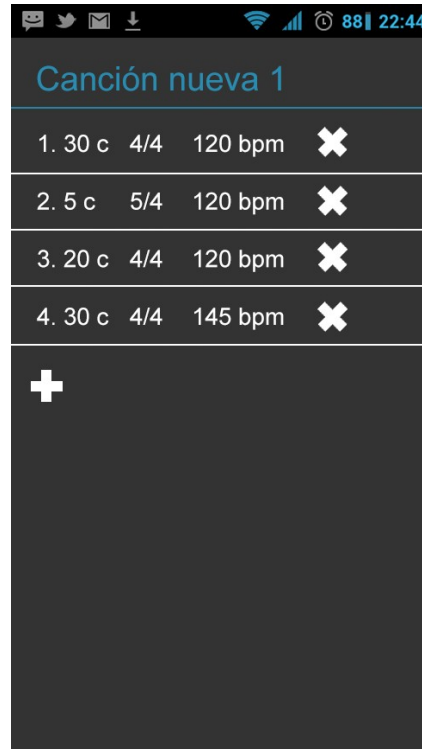


10. Ilustración: Introducción de título de canción



11. Ilustración: Introducción de secuencia

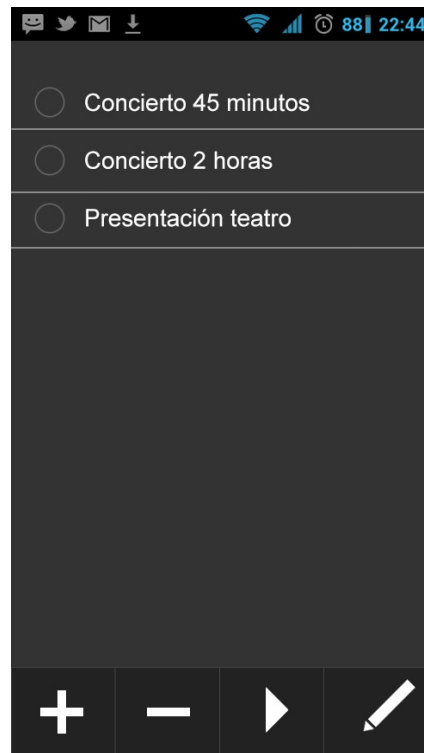
La edición de canciones presentará el siguiente aspecto. La información que se ofrece por cada secuencia es la duración en compases, el tipo de compás y el tempo. Cada secuencia se podrá eliminar pulsando en el icono lateral o modificar manteniéndola pulsada. Pulsando sobre el icono inferior se podrá añadir una secuencia nueva.



12. Ilustración: Edición de canciones

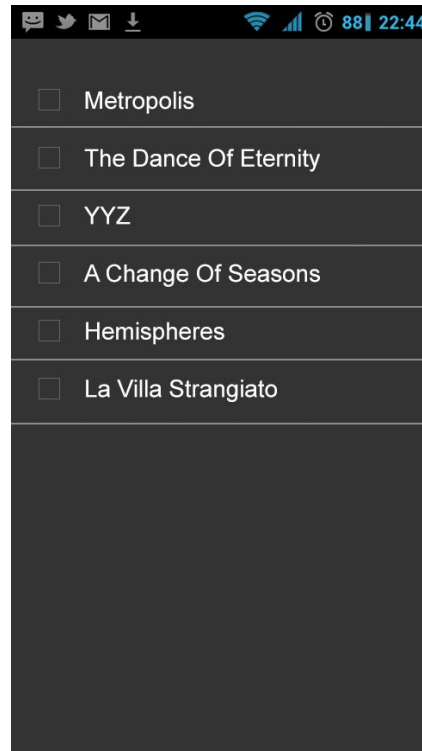
2.3.3 Listas de reproducción

Desde la opción *setlist* se accederá a la reproducción, creación y edición de listas de reproducción (o *setlists*, como se denominan comúnmente en el mundo de la música el listado de canciones a interpretar en un concierto). Primeramente aparecerán las listas de reproducción guardadas (o vacío si no hay ninguna) y los mismos botones en la parte inferior para crear, eliminar, reproducir o editar que en la sección de canciones.



13. Ilustración: Relación de listas de reproducción

A partir de aquí el funcionamiento es muy similar a la creación de canciones mediante los botones de la parte inferior. A la hora de crear una lista de reproducción nueva, después de introducir el título se deben seleccionar las canciones de una lista.



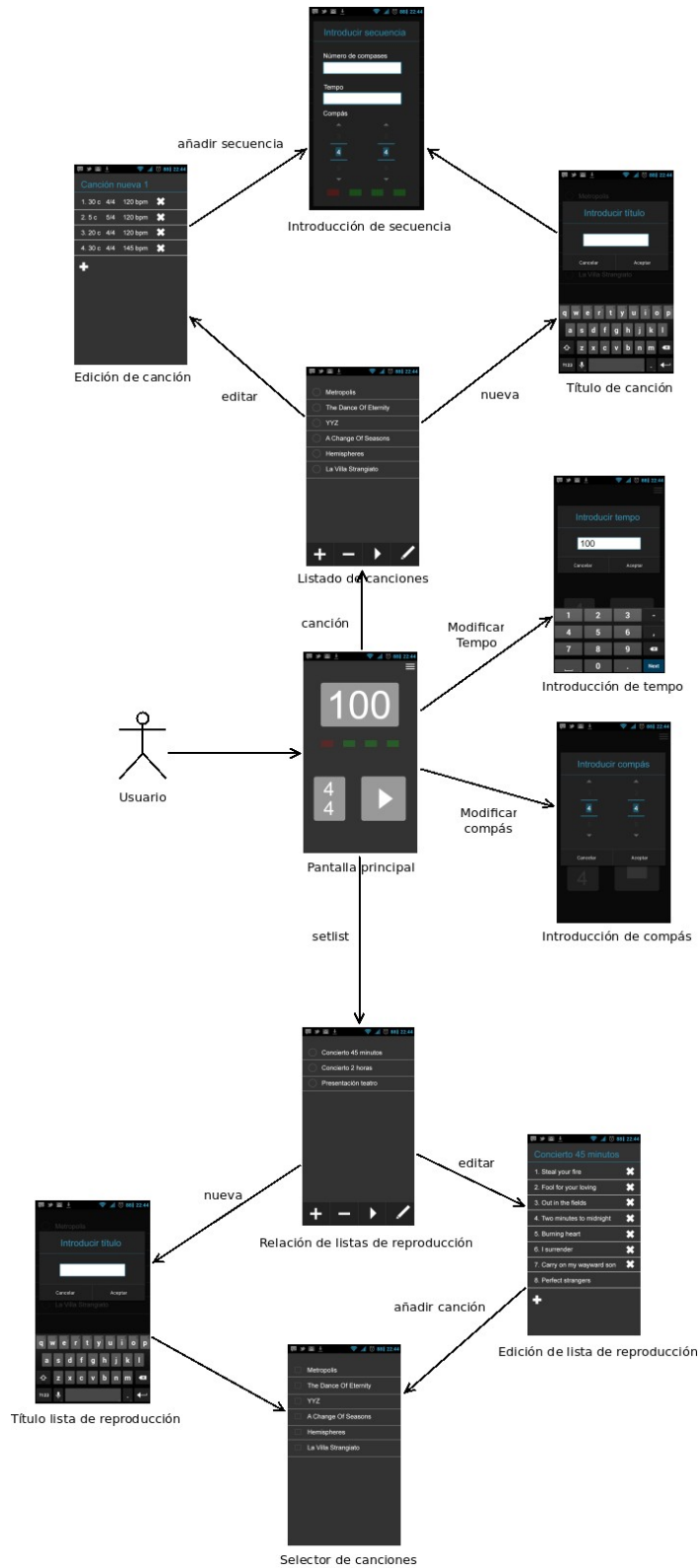
14. Ilustración: Listado de selección de canciones

Pulsando en el botón de edición, aparecerá el listado de canciones de la lista de reproducción seleccionada. Desde aquí se podrá reordenar el *setlist*, añadir canciones nuevas, o eliminar alguna ya existente. Esta última acción no borrará la canción, si no que la descartará en la lista de reproducción. Para reordenar el setlist se mantendrá pulsada una canción y se arrastrará a la posición deseada.



15. Ilustración: Edición de lista de reproducción

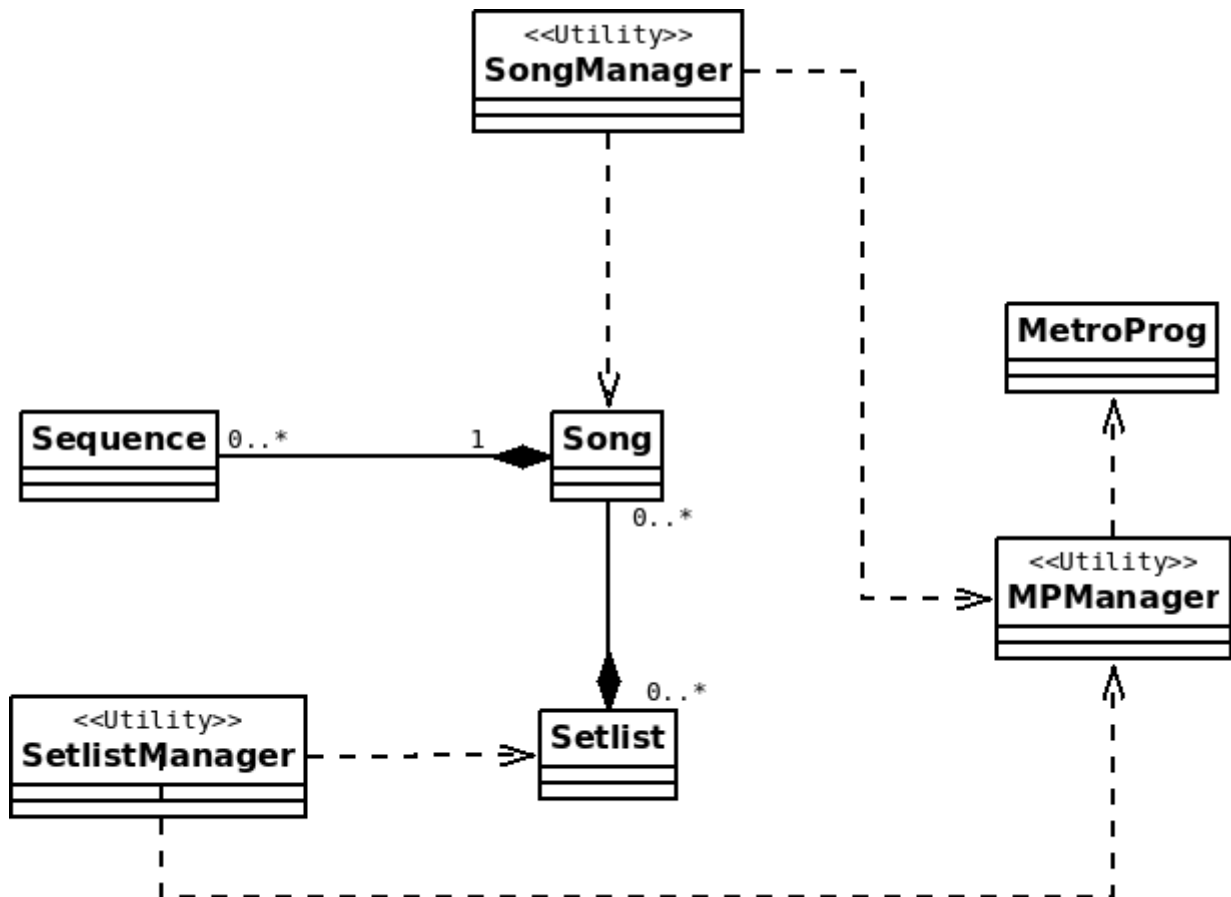
2.3.4 Secuencia de pantallas



2.4 Diseño de arquitectura

Javascript no es un lenguaje orientado a objetos, aunque sí dispone de características de la Programación Orientada a Objetos⁶. Es por ello, que se puede realizar un diseño de arquitectura de forma análoga a cualquier otro lenguaje orientado a objetos.

A continuación se muestra el sencillo diagrama que compone las clases de la aplicación.



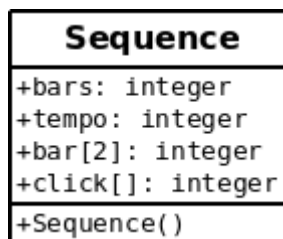
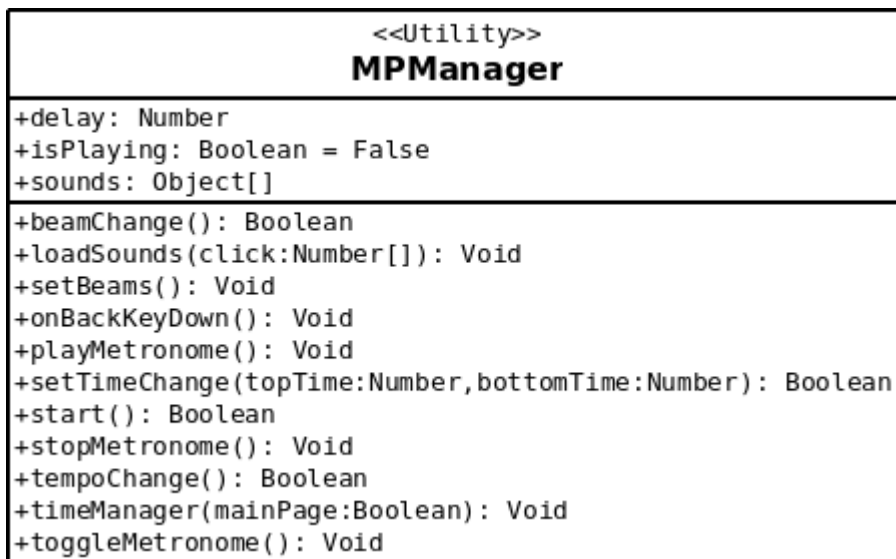
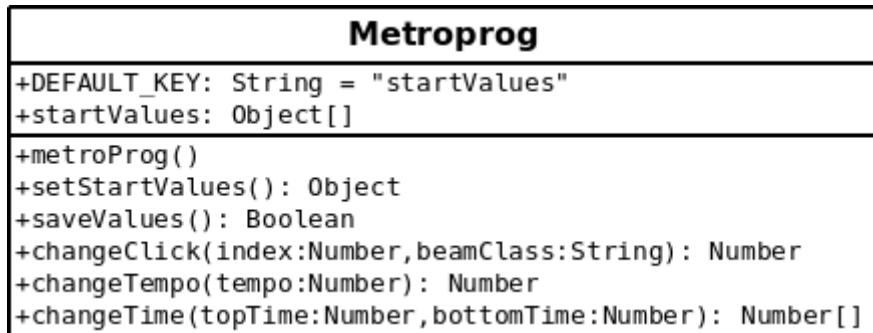
Las canciones están compuestas de secuencias en las que se indica el número de compases, el tempo, el compás y qué pulsos se marcarán como fuertes, débiles o en silencio. Cada secuencia será única y no podrá pertenecer a más de una canción. Las canciones podrán estar vacías.

Los setlist están compuestos de canciones y cada canción puede estar en más de un setlist. Los setlist podrán estar vacíos.

Las clases <<Utility>> son clases estáticas que enlazan con la capa de presentación.

⁶ https://developer.mozilla.org/en-US/docs/Web/JavaScript/Introduction_to_Object-Oriented_JavaScript

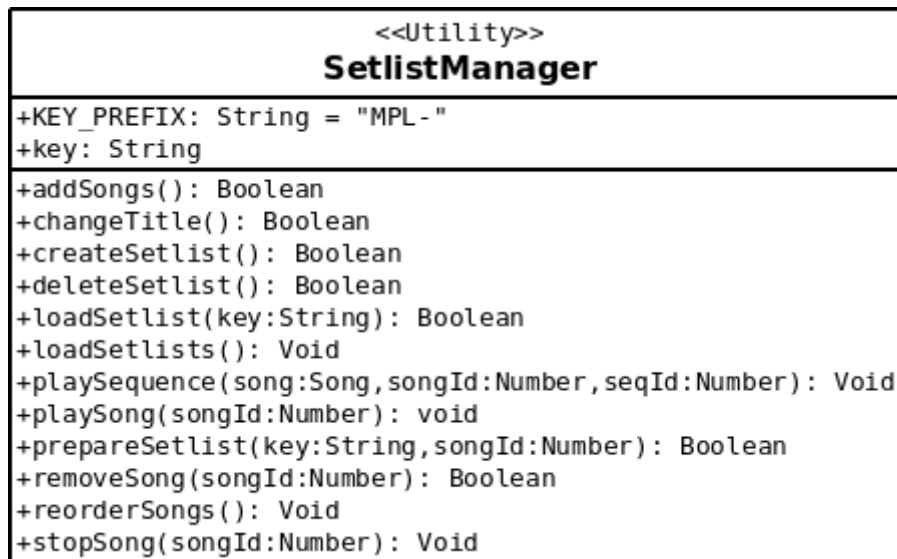
2.4.1 Diagrama detallado



Song
+name: String
+Song() +deleteSequence(songKey:String,seqId:Number): Boolean +get(): Boolean +getSequence(seqId:Number): Sequence +save(key:String): Boolean +setSequence(key:String,pos:Number,seq:Sequence): Boolean +setSong(key:String,name:String): Song +setTitle(key:String,title:String): Boolean

<<Utility>> SongManager
+KEY_PREFIX: String = "MPS-" +delay: Number +isPlaying: Boolean = false +key: String
+changeTitle(): Void +createSong(): Boolean +deleteSequence(seqId:Number): Boolean +deleteSong(): Boolean +loadSequence(seqId:Number): Void +loadSong(key:String): Boolean +loadSongs(container:String): Void +playSong(interval:Number,bar:Number[],seqLength:Number,click:Number[],seqId:Number): Void +prepareSong(seqId:Number): Boolean +setSequence(seqId:Number): Boolean +stopSong(): Void

Setlist
+name: String
+Setlist() +deleteSong(setlistKey:String,songId:Number): Boolean +get(key:String): Boolean +reorder(setlistKey:String,oldPos:Number,newPos:Number): Boolean +save(key:String): Boolean +setSetlist(key:String,name:String): Setlist +setSongs(setlistKey:String,songKeys:String[]): Boolean +setTitle(key:String,title:String): Boolean



2.5 Persistencia

La persistencia de datos se hará mediante la API de HTML5 Web Storage⁷, que mediante su atributo `localStorage` nos permite almacenar pares clave – valor en el dispositivo de manera sencilla, lo cual es suficiente para esta aplicación.

Los valores a guardar serán objetos JSON con la información correspondiente a cada canción. Las claves para recuperar canciones serán nombres del tipo “MPS-nombre”

La estructura de los valores de las canciones será como se describe a continuación:

- **name.** De tipo String será el nombre de la canción.
- **sequences.** Array de objetos JSON que guarda las diferentes secuencias de las canciones con la siguiente estructura:
 - **bars.** Número de compases de la secuencia.
 - **tempo.** Tempo de la secuencia.
 - **bar.** Array de dos elementos que indican numerador [0] y denominador [1] del compás.
 - **click.** Array con tantos elementos como indique el elemento 0 de **bar**. Cada elemento podrá tomar los valores 1, 0 y -1 e indicarán respectivamente pulso fuerte, pulso débil o silencio.

```
{
  "name": String,
  "sequences":
  [
    {
      "bars": Number,
```

⁷ <http://dev.w3.org/html5/webstorage/>


```
        "tempo": Number,  
        "bar": [Number, Number],  
        "click": [Number, ...]  
    }  
]  
}
```

En el caso de las listas de reproducción, las claves serán del tipo “MPL-nombre”. La estructura de los objetos JSON de cada valor será la siguiente:

- **name.** Nombre proporcionado por el usuario para el setlist.
- **songs.** Array que contendrá las claves de cada canción.

```
{  
    "name": String,  
    "songs": [String, ...]  
}
```

Dado que las canciones que componen un *setlist*, a diferencia de las secuencias que forman una canción, sí pueden formar parte de más de un *setlist*, no se incluyen en el propio fichero de *setlists*, tan sólo se hace referencia a ellas mediante su identificador.

Adicionalmente, se añaden los valores de la última sesión del usuario. En ella se almacenan el tempo, el compás y la secuencia de *clicks*, que se recuperarán en la siguiente sesión.

```
{  
    "tempo": Number,  
    "bar": [Number, Number],  
    "click": [Number, ...]  
}
```

2.6 Pruebas previas

2.6.1 Descripción

A los usuarios elegidos para las pruebas se les presentó un prototipo no funcional en HTML para utilizar con un navegador web estándar y una breve descripción de las funcionalidades de la aplicación. En el prototipo pueden acceder a las diferentes opciones y funcionalidades que ofrece la aplicación. El prototipo estaba acompañado de un

pequeño cuestionario al que deberían responder⁸.

2.6.2 Resultados

En líneas generales, el prototipo presentado ha sido evaluado muy positivamente por los usuarios, destacando todos ellos la simplicidad con la que pueden ejecutar las distintas funcionalidades, especialmente las más básicas como cambio de tempo y compás.

Los usuarios 2 y 3 hacen referencia a la posibilidad de subdividir compases de forma personalizada. Esta funcionalidad está prevista (caso de uso 2.1.3) pero no se implementó en el prototipo.

El usuario 1 menciona las dudas a la hora de editar una secuencia. Inicialmente, y tal como se puede observar en los casos de uso, esta funcionalidad no estaba prevista. Se hace necesario por tanto incorporar un segundo botón para la edición de secuencias.

Las ayudas de voz que menciona el usuario 1 requerirían de otros procesos como la grabación de audio, pero donde reside la complejidad de esta funcionalidad está en colocar de forma precisa las grabaciones dentro de la secuencia correspondiente. No se baraja esta posibilidad al menos en la primera versión de la aplicación, pero se tendrá en cuenta para su posible implementación en futuras versiones.

Las opciones de carátula que menciona el usuario 3 hace que la selección de canciones sea más intuitiva, pudiendo personalizar cada una de ellas de la misma forma que un reproductor de audio. Esto implicaría la creación de imágenes y su asociación por parte del usuario, o la inclusión de ciertas imágenes predefinidas en la aplicación. Desde mi punto de vista, esta funcionalidad es más propia de un reproductor multimedia que de un metrónomo.

⁸ Ver anexo

3 Implementación

En el proceso de implementación se han utilizado las siguientes herramientas:

- Apache Cordova
- Adobe PhoneGap
- Apache Ripple Emulator
- Android SDK
- jQuery Mobile
- NetBeans IDE

3.1 Estructura

La aplicación se basa en un único fichero html que se divide en diferentes páginas y diálogos que gestiona jQuery Mobile. Su presentación se descarga en los ficheros CSS de jQuery Mobile y un fichero CSS adicional que se encarga de presentar la pantalla principal y pequeñas adicionales.

Existen cuatro clases Javascript (MetroProg.js, Sequence.js, Song.js, Setlist.js) que se encargan de gestionar el almacenamiento de los datos y tres clases “estáticas” (MPManager.js, SongManager.js y SetlistManager.js) que hacen de puente entre aquéllas y la capa visual gestionando la lógica de la aplicación. Además, se ha creado un fichero main.js que carga los eventos necesarios para el funcionamiento al inicio de la aplicación.

MetroProg.js

Gestiona los valores del metrónomo simple -tempo, compás y subdivisión o “clicks”- y los guarda o recupera del almacenamiento interno para iniciar la siguiente sesión con los últimos utilizados.

MPManager.js

Se encarga de recoger los valores introducidos por el usuario y comunicarse con MetroProg.js. Además gestiona el funcionamiento “play/stop” del metrónomo y los sonidos.

Sequence.js

Es una simple *bean* que contiene los datos de una secuencia de una canción: tempo, duración, compás y *click*.

Song.js

Es la encargada de gestionar el apartado “canciones”, realizando el CRUD de las mismas.

SongManager.js

Se ocupa de la lógica del apartado “canciones”. Recoge los eventos del usuario, gestiona la presentación y se comunica con Song.js para gestionar los datos.

Setlist.js

Esta clase es la que gestiona el CRUD de los *setlists* o listas de reproducción al igual que Song.js

SetlistManager.js

Al igual que SongManager.js gestiona la presentación y se comunica con Setlist.js para efectuar los cambios pertinentes en el almacenamiento.

Debido a la falta de tiempo no he podido hacer una nueva revisión del código profundizando más en las posibilidades que Javascript puede ofrecer en cuanto al desarrollo orientado a objetos. Comparando las clases Song.js y Setlist.js se puede observar que existen varios métodos prácticamente idénticos. Estas clases podrían heredar de otra más genérica en función de las posibilidades que Javascript nos ofrezca en este aspecto haciendo así la aplicación más mantenible.

3.2 Problemas de rendimiento

He encontrado bastantes problemas en el funcionamiento consistente de la aplicación. La función de un metrónomo es marcar el tempo de manera estable, y no he alcanzado este objetivo.

Primeramente traté de utilizar la API Audio de HTML5 para reproducir los sonidos de los pulsos fuertes y débiles. Las pruebas en un navegador estándar, así como en el emulador Apache Ripple⁹ dieron unos resultados parcialmente satisfactorios, esto es, el tiempo era consistente, pero al trabajar con tempos rápidos, algunos pulsos no se reproducían. En las pruebas con un dispositivo real no se reproducía ningún sonido, lo que me hizo descartar esta API.

Posteriormente utilicé el plugin Media¹⁰ de Cordova concebido precisamente para solucionar problemas de reproducción de archivos multimedia en las diferentes plataformas móviles. En esta ocasión los sonidos sí se reproducían en el dispositivo real, pero detecté una notable falta de consistencia en la reproducción del pulso. Además, se reproducía el mismo problema que en las pruebas con la API Audio de HTML5 en el navegador: en tempos rápidos no se reproducían algunos pulsos.

El problema de la inconsistencia en el tiempo podría venir derivado por el hecho de que el plugin Media de Cordova no se carga en memoria hasta que se llama al método *play()*, a pesar de disponer de un constructor, del que a priori esperamos que lo cargue en memoria para que su posterior reproducción sea lo más rápida posible.

Estos problemas me llevaron a buscar alternativas, encontrando una solución parcial en la biblioteca Low Latency Audio¹¹ de Andrew Trice. En el enlace aportado explica la

9 <http://ripple.incubator.apache.org/>

10 <https://github.com/apache/cordova-plugin-media/blob/master/doc/index.md>

11 <http://www.tricedesigns.com/2012/01/25/low-latency-polyphonic-audio-in-phonegap/>

problemática de la precarga en memoria de los archivos de audio:

*In HTML5, you can auto-load a sound so that it is ready when you need it, but don't expect to play more than one at a time. With the PhoneGap Media class, **the audio file isn't actually requested until you invoke "play"**. This occurs inside "[startPlaying](#)" on Android, and "[play](#)" on iOS. What I wanted was a way to preload the audio so that it is immediately ready for use at the time it is needed.*

Con la utilización de esta biblioteca se solucionó el problema de los pulsos que se perdían en tempos rápidos, pero no acabó de mejorarse la consistencia en la reproducción exacta de los pulsos. Para tratar de paliar esta deficiencia, el tiempo calculado se autocorrigie en cada iteración calculando el tiempo de proceso de la función que reproduce los sonidos:

```
playMetronome : function ( )
{
    var MINUTE = 60000;           // Minuto en milisegundos.
    var tempo = MPManger.mp.startValues.tempo *
        MPManger.mp.startValues.bar[1] / 4; // Tempo adaptado a las figuras.
    var interval = MINUTE / tempo; // Tempo efectivo.
    var barLength = MPManger.mp.startValues.bar[0]; // Longitud del compás.
    var click = MPManger.mp.startValues.click; // Subdivisión del compás.
    var barPosition = 0; // Posición del compás.
    var start; // Momento de inicio de la función.
    var end; // Momento final de la función.
    var newInterval; // Intervalo ajustado.

    MPManger.isPlaying = true;

    function play()
    {
        // Se recoge el tiempo inicial
        start = Date.now();

        // Se "apagan" las luces que pudieran estar encendidas.
        $('li.beams ul li a').removeClass('beam-on');

        if ( MPManger.isPlaying )
        {
            // Se "enciende" la luz correspondiente.
            $('li.beams ul li:eq(' + barPosition + ') a').addClass('beam-on');
```

```
// Se reproduce el sonido.
switch(click[barPosition])
{
    case 0:
        LowLatencyAudio.play('low');
        break;
    case 1:
        LowLatencyAudio.play('high');
        break;
}
// Avanza la posición del compás
if ( ++barPosition == barLength )
    barPosition = 0;

// Se recoge el tiempo final.
end = Date.now();

// Se calcula la desviación del intervalo.
newInterval = interval - (end - start);

// Se dispara de nuevo la reproducción.
MPManager.delay = window.setTimeout(play, Math.round(newInterval));
}
}
play();
}
```

A pesar de esta corrección, el tempo sigue siendo inconsistente. Se trató de usar la API de Javascript Performance¹² para dotar al temporizador de mayor precisión, pero no era reconocida por Cordova. De cualquier forma no supondría mayor ventaja, dado que la precisión en el método `setTimeout` es de milisegundos.

Finalmente, el intervalo calculado es redondeado, puesto que `setTimeout` trunca internamente los valores decimales¹³.

En cualquier caso, la utilización de Javascript / Cordova para este tipo de aplicaciones que requieren de una especial solidez y precisión no parece ser lo más adecuado, dado que los intervalos de las funciones temporizadoras pueden verse afectados por otras tareas del sistema operativo, como explican en la web para desarrolladores de Mozilla¹⁴:

12 <https://developer.mozilla.org/en-US/docs/Web/API/Performance.now%28%29>

13 <http://stackoverflow.com/questions/8468790/is-it-safe-to-pass-setinterval-or-settimeout-a-fractional-delay>

14 <https://developer.mozilla.org/en-US/docs/Web/API/window.setTimeout#Notes>

In addition to "clamping", the timeout can also fire later when the page (or the OS/browser itself) is busy with other tasks.

La biblioteca LowLatencyAudio, a pesar de mejorar considerablemente la API Media de Cordova, presenta otro problema en la compilación / ejecución. Al iniciar la aplicación se ejecutan correctamente los callbacks de ejecución correcta del método preloadFX que es el encargado de precargar los sonidos, pero en muchas ocasiones los sonidos no llegan a reproducirse. Se utilizan dos sonidos (archivos .wav de menos de 1KB), uno para marcar los tiempos fuertes y otro para los débiles. Unas veces carga un solo sonido, otras los dos y otras ninguno. Las pruebas con esta biblioteca se realizaron ya entrado el mes de junio, por lo que no me ha dado tiempo a investigar si se trata de un problema en tiempo de compilación (algo poco probable, puesto que la salida del log de NetBeans no muestra ningún error) o en tiempo de ejecución. He probado a poner callbacks de éxito en el método play, pero a pesar de ejecutarse, los sonidos siguen sin reproducirse. Por estos motivos **se hace necesario por el momento compilar y ejecutar la aplicación hasta que se reproduzcan correctamente los sonidos.**

3.3 Aspectos pendientes de finalización

A pesar de que se presenta una aplicación prácticamente finalizada, debido al límite de tiempo, han sido algunos los aspectos que no se han podido finalizar.

Internacionalización

Los textos literales están "hardcoded", queda pendiente crear un sistema que permita guardar los textos en diferentes idiomas y cargar el que seleccione el usuario.

Splash screen

Al inicio de la aplicación se deben cargar los archivos de audio. Idealmente este proceso se debe realizar mientras se muestra una *splash screen* o pantalla de inicio. Por el momento, **al iniciar la aplicación se debe esperar para usarla a que aparezcan dos avisos 'OK' al inicio.**

Integración de diálogos

Cordova dispone del plugin *Dialogs*¹⁵ que permite utilizar los cuadros de diálogo nativos del sistema. Actualmente se están utilizando los diálogos por defecto.

Ordenación de listas de reproducción

Está previsto que las listas de reproducción se puedan reordenar fácilmente pulsando y arrastrando o algún otro método. Esta funcionalidad no está implementada.

Orientación de la pantalla

No está definido el diseño para orientación horizontal de la pantalla. Tampoco está

¹⁵ <https://github.com/apache/cordova-plugin-dialogs/blob/master/doc/index.md>

bloqueada la orientación vertical.

Bloqueo de pantalla

No está controlada la entrada en reposo de la pantalla.

Disposición de las “luces”

Se debe buscar un método para distribuir de forma correcta las “luces” de los pulsos.

Deficiencias visuales

Se produce un pequeño efecto de “flash” al ir de una pantalla a otra cuya causa aún no ha sido identificada

4 Conclusiones

HTML5, CSS y Javascript son una excelente puerta de entrada al desarrollo de aplicaciones para dispositivos móviles, puesto que son unos lenguajes muy extendidos y con una curva de aprendizaje bastante liviana. En mi caso, al haber trabajado como desarrollador web front-end, me resultó muy natural trabajar con estos lenguajes apoyándome en el framework jQuery Mobile.

Al igual que en el desarrollo web front-end, donde se trabaja para navegadores de distintos fabricantes y que funcionan sobre diversos sistemas operativos (y en los últimos tiempos para todo tipo de dispositivos móviles), nos enfrentamos también a la dificultad de desarrollar una única vez para muchas plataformas y dispositivos diferentes, con la salvedad de que podemos limitarnos a las que deseemos. A pesar de todo, la diversidad de *gadgets* que funcionan con el mismo sistema operativo hacen de esta tarea un reto.

El desarrollo con PhoneGap / Cordova es bastante rápido, pero desde la experiencia de este trabajo lo reservaría únicamente para aplicaciones que no fueran muy exigentes en el rendimiento ni necesiten trabajar a bajo nivel.

El objetivo que me planteé al inicio del semestre fue el de crear una aplicación comercial, plenamente funcional que se pudiera publicar en las diferentes plataformas de aplicaciones, y he de decir que no se ha alcanzado. Pero esta experiencia sí me ha servido por una parte para -evidentemente- conocer el desarrollo en HTML5 para móviles y por otra para animarme a portar el código a los lenguajes nativos de algunas plataformas, algo que sin duda haré en el futuro.

ANEXO

Usuario 1

¿Te parece intuitivo el cambio de tempo y de compás?

Sí, me parece muy intuitivo. En el mismo instante de ver la pantalla, sabes cómo hacer los cambios de tempo y compás.

¿Has podido seguir los pasos de creación de canción?

Perfectamente.

¿Y la edición?

Sin problema.

¿Te ha parecido intuitivo el proceso?

Mucho. Únicamente, si necesitas modificar una secuencia de una canción, no sabría cómo seleccionar dicha secuencia. Es decir, pulsar sobre ella, mantener el ratón pulsado, doble click...

¿Has podido seguir los pasos de creación de *setlist*?

Sí.

¿Y la edición?

Perfectamente.

¿Te ha parecido intuitivo el proceso?

Mucho. Sólo estaría lo dicho anteriormente.

¿Echas en falta alguna funcionalidad?

Sería muy útil poder insertar indicaciones de audio del tipo: "Percusión en 1,2,3,4"; "voz en 1,2"...es decir entradas de instrumentos o indicaciones de cambios de tempo,etc.

¿Consideras útil la aplicación?

Muy útil. Para los músicos tener la claqueta de una canción entera, con sus cambios de tempo y compás, resulta utilísimo; para ensayar y tocar la canción, hacer nuevos arreglos y tocarlos encima del tempo correcto...Si además se puede hacer con una aplicación de móvil, en cualquier momento y lugar...¡estupendo!

En general ¿qué opinión te merece la aplicación?

Muy útil e intuitiva. Añade la posibilidad de construir la claqueta de canciones o set list enteros.

Usuario 2

¿Te parece intuitivo el cambio de tempo y de compás?

Sí, tocar y listo.

¿Has podido seguir los pasos de creación de canción?

Sí, muy sencillo

¿Y la edición?

Sí

¿Te ha parecido intuitivo el proceso?

Sí

¿Has podido seguir los pasos de edición de *setlist*?

Sí

¿Y la edición?

Sí

¿Te ha parecido intuitivo el proceso?

Sí

¿Echas en falta alguna funcionalidad?

Sí, poder hacer subdivisiones dentro del propio compás. Si necesito entrenar con un 10/8 me gustaría poder agrupar las corcheas a mi manera; 3-3-2-2; 3-2-2-3;....

¿Consideras útil la aplicación?

Sí

En general ¿qué opinión te merece la aplicación?

Fácil de manejar, muy útil para los músicos... Un 9/10

Usuario 3

¿Te parece intuitivo el cambio de tempo y de compás?

Sí, no hay opción a duda

¿Has podido seguir los pasos de creación de canción?

Sí

¿Y la edición?

Sí

¿Te ha parecido intuitivo el proceso?

Mucho

¿Has podido seguir los pasos de edición de *setlist*?

Sí

¿Y la edición?

Sí

¿Te ha parecido intuitivo el proceso?

Sí

¿Echas en falta alguna funcionalidad?

A lo mejor una opción de carátula para hacer la búsqueda más visual y las opciones personalizar la subdivisión del compás (esto la haría más profesional en comparación a otras).

¿Consideras útil la aplicación?

Sí

En general ¿qué opinión te merece la aplicación?

Respeto la tendencia del mercado a centrarse en apps más operativas y menos estéticas. Yo la usaría frente a otras partiendo de su interface.