

gastar group

---

**Reproductor de Música**

**Configuration Managment Plan**  
**(CM\_Plan)**

---

**Autores:**  
**<Gasparini, Roman;**  
**Tarres, Martin>**

**Document Version: 1.0.0**

## Historial de Revisiones

Version	Fecha	Resumen de Cambios	
1.0.0	22/06/2017	Primera versión	

## **Tabla de contenidos**

- 1 Integrantes**
- 2 Release Notes**
- 3 Manejo de las Configuraciones**
- 4 Requerimientos**
- 5 Arquitectura**
- 6 Diseño e Implementación**
- 7 Pruebas unitarias y del sistemas**
- 8 Datos históricos**
- 9 Información Adicional**

## 1 Integrantes

El grupo Gastar group está conformado por

<b>Gasparini, Roman</b>	<b>36587720</b>	<b>roman.gasparini@gmail.com</b>
-------------------------	-----------------	----------------------------------

<b>Tarres, Martin</b>
-----------------------

## 2 Release Notes

El documento de las notas de entregas se puede encontrar en la siguiente URL.

## 3 Manejo de las Configuraciones

El plan de gestión de las configuraciones se encuentra en la siguiente dirección, URL.

## 4 Requerimientos

El documento de Especificación de Requerimientos de Software utilizado en el proyecto se encuentra en la siguiente URL.

## 5 Arquitectura

### Diagrama de Arquitectura (Diagrama)

En el diagrama anterior podemos observar la arquitectura del sistema, en el cual utilizamos el mas indicado de los patrones para este tipo de proyectos, como lo es el reproductor de música, que es el MVC. Donde separa la interfaz de usuario (vista), de la lógica, es decir del modelo, y del controlador que es el que interpreta las acciones. Como este patrón dispone de un bajo acoplamiento, nos permite poder cambiar de vistas con facilidad, como pasar de la lista principal, a la vista secundaria, es decir de la carpeta contenedora de las pistas de audio, a la vista de la lista de reproducción. De este modo, el modelo puede ser reutilizable por cada vista, y así cada una de estas puede implementar funcionalidades totalmente distintas del modelo.

También debido a su bajo acoplamiento, es posible hacer nuevas implementaciones, o cambiar la vista, de manera muy sencilla.

En el diagrama podemos observar que las líneas continuas (—▶) como el flujo principal, donde el usuario tiene interacción con la interfaz gráfica de usuario, el controlador es notificado sobre los cambios, y la orden de este ultimo al modelo.

Las líneas punteadas (- - - ▶) indican el flujo secundario, que hace referencia a la actualización de las vistas.

## 6 Diseño e Implementación

En el siguiente apartado mostraremos los diagramas que conforman el diseño del proyecto.

A continuación podemos observar el diagramas de paquetes, este esta compuesto por los paquetes para cada tipo de clases, y las librerías que estos utilizan. Dentro del paquete vistas se encuentran las clases Vista, VistaListaReproduccion y VistaPrincipal.

(Diagrama de paquetes)

## Diagrama de clases

La siguiente imagen nos muestra el diagrama de clases, el cual contendrá las clases, Main, la clase principal, modelo, vista, Observador, vistaListaReproduccion, vistaPrincipal.

## **7 Pruebas Unitarias y del Sistema**

## **8 Datos Históricos**

## **9 Información adicional**











