

Instituto Tecnológico de Costa Rica
Escuela de Ingeniería en Computadores

TestPlan - MusicBox

Algoritmos y Estructuras de Datos II (CE 1103)
Verano 2025–2026

Estudiante:

Janik Zúñiga Hamilton

Profesor:

Leonardo Andres Araya Martinez

Fecha:

13 de enero de 2026

Índice

1. Introducción	2
2. Objetivo del Plan de Pruebas	2
3. Alcance del Plan de Pruebas	2
4. Estrategia de Pruebas	3
5. Entorno de Pruebas	3
6. Identificación de Casos de Prueba	4
7. Criterios de Aceptación	5
8. Resultados de la Ejecución de Pruebas	5
9. Conclusiones	5

1. Introducción

El presente documento corresponde al **Plan de Pruebas (Test Plan)** del proyecto **MusicBox**, desarrollado como parte de un ejercicio académico en el área de Ingeniería en Computación. Este documento tiene como finalidad describir, de manera estructurada y formal, el conjunto de actividades de verificación y validación realizadas sobre el sistema, con el objetivo de asegurar que el software cumple con los requerimientos funcionales y de calidad establecidos.

La redacción se presenta de forma estrictamente impersonal, siguiendo las buenas prácticas de documentación técnica, y está orientada a servir como evidencia objetiva del proceso de aseguramiento de la calidad del software. El contenido aquí expuesto no describe detalles internos de implementación, sino que se enfoca en el comportamiento observable del sistema desde el punto de vista de las pruebas.

2. Objetivo del Plan de Pruebas

El objetivo principal de este plan de pruebas es definir y documentar el proceso mediante el cual se verifica el correcto funcionamiento del sistema MusicBox. En particular, se busca comprobar que el sistema:

- Cumple de manera íntegra con los requerimientos funcionales definidos para el proyecto.
- Procesa correctamente entradas válidas e inválidas relacionadas con partituras musicales.
- Calcula de forma precisa las frecuencias y duraciones asociadas a cada nota musical.
- Permite la reproducción de la secuencia musical en ambos sentidos de recorrido.
- Presenta un comportamiento estable y predecible ante diferentes escenarios de uso.
- Se encuentra respaldado por pruebas automatizadas que facilitan su mantenimiento.

3. Alcance del Plan de Pruebas

El alcance de este plan de pruebas abarca la validación de los componentes lógicos del sistema MusicBox. Las pruebas se centran en verificar el comportamiento

funcional de los módulos que conforman la aplicación, sin considerar aspectos relacionados con interfaces gráficas ni pruebas de rendimiento intensivo.

Los componentes incluidos dentro del alcance son:

- El módulo de interpretación y validación de partituras musicales.
- El control de tempo y el cálculo proporcional de duraciones.
- La estructura de datos basada en una lista doblemente enlazada.
- El módulo de reproducción musical en modo normal y reverso.
- El mecanismo de visualización de información en modo verbose.

Quedan explícitamente fuera del alcance las pruebas de rendimiento, pruebas de estrés y pruebas de interfaz gráfica, dado que el sistema opera mediante una interfaz de línea de comandos.

4. Estrategia de Pruebas

Para la validación del sistema se adoptó una estrategia basada en **pruebas unitarias automatizadas**. Cada prueba se diseñó para evaluar un comportamiento específico del sistema de manera aislada, reduciendo la posibilidad de interferencias entre componentes y facilitando la identificación de errores.

Las pruebas fueron implementadas utilizando el framework **xUnit**, ampliamente utilizado en el ecosistema .NET, lo que permite una ejecución reproducible y automatizada del conjunto de pruebas.

La ejecución de las pruebas se realiza mediante el siguiente comando:

```
dotnet test MusicBox.sln
```

5. Entorno de Pruebas

Las pruebas se ejecutaron bajo las siguientes condiciones técnicas:

- Lenguaje de programación: C#.
- Plataforma de desarrollo: .NET.
- Sistema operativo: Windows 10 Pro 22H2.
- Framework de pruebas: xUnit.

- Entorno de ejecución: Consola de comandos.

Este entorno representa de forma adecuada el contexto esperado de uso del sistema durante su evaluación académica.

6. Identificación de Casos de Prueba

Los casos de prueba fueron identificados y diseñados de acuerdo con los requerimientos funcionales del sistema. Cada caso de prueba valida un aspecto concreto del comportamiento del software, incluyendo tanto escenarios nominales como situaciones de error.

ID	Componente	Descripción del Caso de Prueba	Resultado Esperado
CP-01	Parser	Procesamiento de una partitura con formato válido	Lista generada sin errores
CP-02	Parser	Detección de formato inválido en la partitura	Excepción controlada
CP-03	Tempo	Validación de valores límite del tempo	Valor aceptado o excepción
CP-04	Estructuras	Recorrido forward de la lista musical	Orden correcto de notas
CP-05	Estructuras	Recorrido backward de la lista musical	Orden inverso correcto
CP-06	Playback	Reproducción secuencial hacia adelante	Notas reproducidas en orden
CP-07	Playback	Reproducción secuencial hacia atrás	Notas reproducidas en orden inverso
CP-08	Verbose	Visualización de frecuencia y duración	Información mostrada correctamente
CP-09	Limpieza	Eliminación de la partitura cargada	Lista vacía
CP-10	Integración	Ejecución completa del comando demo	Reproducción sin errores

Cuadro 1: Casos de prueba principales del sistema MusicBox

7. Criterios de Aceptación

El sistema MusicBox se considera aceptado cuando se cumplen de manera simultánea los siguientes criterios:

- Todas las pruebas unitarias definidas se ejecutan de forma exitosa.
- Se supera el número mínimo de pruebas unitarias requerido.
- El sistema compila sin errores ni advertencias críticas.
- El comportamiento observado coincide con los requerimientos funcionales establecidos.

8. Resultados de la Ejecución de Pruebas

Como resultado de la ejecución del plan de pruebas, se obtuvo el siguiente resumen:

- Total de pruebas ejecutadas: 23.
- Pruebas exitosas: 23.
- Pruebas fallidas: 0.
- Pruebas omitidas: 0.

Los resultados obtenidos evidencian que el sistema cumple de manera satisfactoria con los requerimientos definidos, proporcionando un alto grado de confianza en su estabilidad y correcto funcionamiento.

9. Conclusiones

El desarrollo del proyecto MusicBox permitió aplicar de manera integral los principios fundamentales de la ingeniería de software, combinando conceptos de programación orientada a objetos, estructuras de datos, validación de entradas y pruebas automatizadas dentro de un sistema funcional y coherente. A lo largo del proyecto se evidenció la importancia de una correcta separación de responsabilidades, lo cual facilitó tanto el desarrollo como la comprensión y validación del sistema.

La arquitectura modular adoptada demostró ser adecuada para el problema planteado, ya que permitió dividir el sistema en componentes claramente definidos, tales

como la interfaz de usuario, la lógica central, el dominio musical, la estructura de datos y el módulo de reproducción. Esta separación no solo mejora la mantenibilidad del software, sino que también favorece su extensibilidad, permitiendo que futuros cambios o mejoras puedan realizarse sin afectar de manera significativa al resto del sistema.

El uso de una lista doblemente enlazada como estructura de datos principal resultó una decisión técnica justificada, dado que el sistema requiere recorrer la partitura musical tanto en sentido directo como inverso. Esta elección permitió implementar la reproducción forward y backward de forma eficiente y sin necesidad de duplicar o transformar los datos almacenados, cumpliendo así con los requerimientos funcionales del proyecto.

Asimismo, la correcta modelación del dominio musical mediante clases como MusicEvent, Tempo, NoteCatalog y FigureCatalog permitió representar de manera precisa los conceptos involucrados en la reproducción musical. En particular, el control de tempo y el cálculo proporcional de las duraciones de las figuras musicales evidencian un tratamiento riguroso de las reglas del dominio, garantizando coherencia entre la representación interna y el comportamiento observable del sistema.

La implementación de una interfaz de audio (IPlayerAudio) y su correspondiente implementación concreta (BeepAudioEngine) permitió desacoplar la lógica musical del mecanismo específico de reproducción. Este diseño refuerza el principio de inversión de dependencias y facilita la sustitución del motor de audio en futuras versiones del sistema, sin requerir modificaciones en la lógica central.

Desde el punto de vista de la calidad del software, la inclusión de un conjunto amplio de pruebas unitarias automatizadas constituye uno de los aspectos más relevantes del proyecto. La ejecución de 23 pruebas unitarias, superando el mínimo requerido, permitió validar no solo los casos nominales, sino también escenarios límite y situaciones de error. Esto proporciona un alto grado de confianza en la estabilidad y corrección del sistema, además de servir como mecanismo de detección temprana ante posibles regresiones.

El uso de implementaciones simuladas, como FakeAudio, facilitó la realización de pruebas deterministas y controladas, evitando dependencias externas y permitiendo verificar con precisión aspectos críticos como la frecuencia y la duración de las notas reproducidas. Este enfoque demuestra una correcta comprensión de las buenas prácticas en pruebas de software.

Finalmente, el proyecto MusicBox cumple satisfactoriamente con los objetivos planteados, ofreciendo una solución funcional, bien estructurada y adecuadamente validada. El sistema desarrollado no solo resuelve el problema propuesto, sino que

también constituye una base sólida para futuras extensiones, tales como la incorporación de nuevos motores de audio, soporte para más figuras musicales o interfaces de usuario más avanzadas. En conjunto, el proyecto evidencia un correcto dominio de los conceptos técnicos abordados y una aplicación adecuada de metodologías de desarrollo de software.