



Instituto Tecnológico de Costa Rica

Escuela de Ingeniería en Computadores

Algoritmos y Estructuras de Datos I (CE 1103)

Proyecto I: MusicBox

Profesor:

Nombre del Profesor

Estudiantes:

Steven

Alejandro

Verano 2024-2025

Índice

1. Descripción General	2
2. Tecnologías y Herramientas	2
3. Estructura del Proyecto	2
4. Funcionalidades del Menú (consola)	2
5. División de Responsabilidades	3
6. Orden de Implementación	3
7. Control de Versiones	4
8. Reuniones Programadas	4
9. Documentación	4

1. Descripción General

Sistema de reproducción de partituras musicales que implementa una lista doblemente enlazada para almacenar y reproducir notas musicales con duraciones configurables. Desarrollado para el curso de Algoritmos y Estructuras de Datos I del TEC.

2. Tecnologías y Herramientas

- C# (.NET 6+)
- NAudio (biblioteca de audio)
- NUnit (testing)
- Git/GitHub
- Visual Studio 2022
- Overleaf (documentación)

3. Estructura del Proyecto

```
MusicBox/  
  Program.cs          # Punto de entrada, menú principal  
  Models/  
    Note.cs           # Clase para las notas musicales  
    Node.cs           # Nodo para la lista enlazada  
    DoublyLinkedList.cs # Implementación de la lista  
  Services/  
    MusicPlayer.cs    # Lógica de reproducción  
    Parser.cs         # Para parsear el input del usuario  
  Tests/              # Directorio para los tests unitarios
```

4. Funcionalidades del Menú (consola)

1. Ingresar nueva partitura
2. Reproducir hacia adelante
3. Reproducir hacia atrás
4. Configurar duración de negra
5. Mostrar partitura actual
6. Salir

5. División de Responsabilidades

Steven

- DoublyLinkedList completa (ID 002)
- Implementación de reproducción de audio (ID 003)
- Control de velocidad de reproducción (ID 004)
- 5 Tests unitarios documentados

Alejandro

- Parser del formato de entrada (ID 001)
- Reproducción bidireccional (ID 005)
- Sistema de navegación en la lista
- 5 Tests unitarios documentados

6. Orden de Implementación

Steven

1. DoublyLinkedList base
2. Integración NAudio
3. Reproducción básica
4. Control de velocidad
5. Tests unitarios

Alejandro

1. Parser básico
2. Sistema de navegación
3. Reproducción bidireccional
4. Refinamiento del parser
5. Tests unitarios

7. Control de Versiones

- Cada feature se desarrolla en una rama separada
- Pull requests a develop cuando el feature esté completo
- Pull requests requieren tests pasando si es posible

8. Reuniones Programadas

- 10 enero — 7:00 PM: planificación
- 12 enero — 7:00 PM: nose
- 14 enero — 5:00 PM: Revisión

9. Documentación

- Documento compartido en Overleaf
- Cada uno documenta sus tests según ID 007
- Plazo máximo: noche anterior a la entrega
- Incluir screenshots de tests ejecutándose