

Music Box

Instituto Tecnológico de Costa Rica
Escuela de Ingeniería en Computadores
Algoritmos y Estructuras de Datos II (CE 1103)
Verano 2024-2025

Objetivos

General

- Implementar un reproductor de partituras utilizando estructuras de datos lineales

Específicos

- Implementar una lista doblemente enlazada que le permita al usuario reproducir notas musicales especificando únicamente su altura y duración.
- Desarrollar algoritmos para dar solución a un problema.

Teoría musical básica

Propiedades del sonido

Refiérase [a este artículo](#) para entender en detalle las cuatro propiedades básicas del sonido, las cuales son

1. Altura
2. Duración
3. Intensidad
4. Timbre











Para este proyecto se hará uso de las cuatro propiedades del sonido para crear un reproductor musical.

El pentagrama

El pentagrama corresponde a la base de las partituras, y consiste en cinco líneas horizontales y equidistantes, sobre las cuales se escriben las figuras y demás elementos musicales.

Figuras musicales

Las partituras están escritas utilizando varios elementos formales para hacerlo. Uno de sus elementos más básicos son las figuras musicales. La siguiente figura resume el nombre de la figura, su símbolo y su duración. Para este proyecto, la unidad (la negra) tendrá un valor configurable por el usuario; por ejemplo, se puede decir que una negra va a durar un segundo, y el resto de figuras se adecuarán de forma proporcional.

Redonda			4 tiempos
Blanca			2 tiempos
Negra			1 tiempo
Corchea			1/2 tiempo
Semicorchea			1/4 tiempo

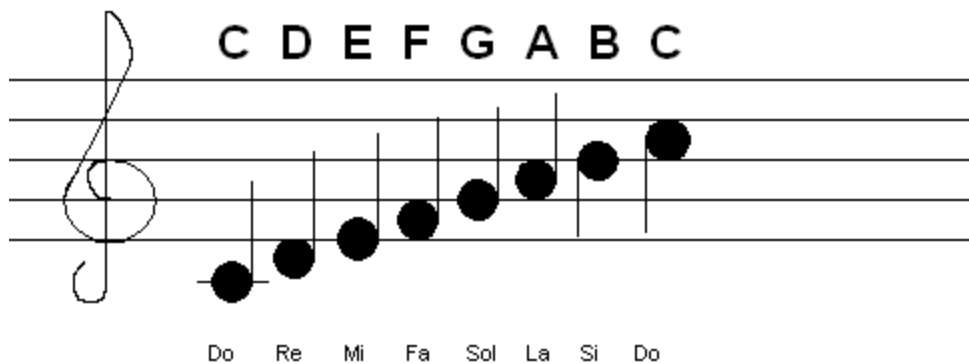
El proyecto debe soportar únicamente las figuras especificadas en la imagen anterior.

Notas musicales

Los nombres de las notas musicales son Do, Re, Mi, Fa, Sol, La y Si. La frecuencia de cada nota ya está predeterminada, y es la siguiente

- **Do (C4):** 261.63 Hz
- **Re (D4):** 293.66 Hz
- **Mi (E4):** 329.63 Hz
- **Fa (F4):** 349.23 Hz
- **Sol (G4):** 392.00 Hz
- **La (A4):** 440.00 Hz
- **Si (B4):** 493.88 Hz

El proyecto requiere que las notas se reproduzcan en la frecuencia exacta, o al menos muy aproximada, para que no se provoquen desafinaciones a causa del reproductor.



Requerimientos

Este proyecto consiste en diseñar e implementar un reproductor de partituras, donde el usuario ingrese su “partitura” de forma escrita y el sistema cree una lista doblemente enlazada que almacene cada una de las notas musicales y su duración.

A continuación, se resumen los requerimientos del proyecto. Esta misma tabla será la rúbrica de evaluación.

ID	DESCRIPCIÓN	PUNTOS
001	<p>El usuario debe ingresar sus partituras como un string usando el siguiente formato</p> <p>(NOTA 1, FIGURA 1), (NOTA 2, FIGURA 2), ... (NOTA n, FIGURA n)</p> <p>Por ejemplo</p> <p>(Do, negra), (Re, blanca), (La, Semicorchea)</p>	15
002	El sistema debe crear una lista doblemente enlazada que almacene cada una de las notas ingresadas por el usuario, así como su duración.	10
003	El sistema debe ser capaz de reproducir la información que contiene la lista doblemente enlazada, reproduciendo un sonido con un timbre pre definido y estático, pero en la frecuencia exacta de la nota en cuestión.	15
004	<p>Es posible modificar el valor en segundos que tarda en reproducirse la figura principal (la negra), y el resto de figuras deben ajustarse proporcionalmente a ese cambio.</p> <p>El rango válido de valores para la negra es [100ms, 5s].</p>	10
005	Es posible reproducir las canciones de adelante hacia atrás o de atrás para adelante.	10

006	Se debe definir al menos 10 tests funcionales, que prueben funcionalidades específicas del código del proyecto.	30
007	La documentación externa corresponderá al <i>test plan</i> donde se especifique lo siguiente <ul style="list-style-type: none"> 1. Qué hace el test 2.Cuál es la entrada que se le ingresa al código probado 3.Cuál es la salida que se espera 4. Un screenshot del test corriendo 	10
Total		100

Aspectos operativos

- El trabajo se realizará en grupos de tres personas.
- El uso de Git y Github es obligatorio.
- La fecha de entrega será según lo especificado en el TEC Digital.
- Se entrega en el TEC digital un link al repositorio donde vive el proyecto.
- Los estudiantes pueden seguir trabajando en el código hasta 15 minutos antes de la cita revisión oficial.

Evaluación

- Los proyectos que no cumplan con los siguientes requisitos no serán revisados:
 - Toda la solución debe estar integrada
 - La interfaz de usuario debe estar implementada e integrada
- El código tendrá un valor total de 80%, la documentación 20%. De estas notas se calculará la *Nota Final del Proyecto*.
- Aún cuando el código y la documentación tienen sus notas por separado, se aplican las siguientes restricciones
 - **Si no se entrega documentación en formato PDF, automáticamente se obtiene una nota de 0.**
 - Si no se utiliza un manejador de código se obtiene una nota de 0.
 - Si la documentación no se entrega en la fecha indicada se obtiene una nota de 0.
 - El código debe desarrollarse en Arduino programming language, si no, se obtendrá una nota de 0.
- Cada estudiante tendrá 15 minutos para exponer su trabajo al profesor y defenderlo, es responsabilidad de los estudiantes mostrar todo el trabajo realizado, por lo que se recomienda tener todo listo antes de entrar a la defensa.
- Cada grupo es responsable de llevar los equipos requeridos para la revisión, si no cuentan con estos deberán avisar al menos 2 días antes de la revisión a el profesor para coordinar el préstamo de estos.

- Durante la revisión únicamente podrán participar los miembros del grupo, asistentes, otros profesores y el coordinador del área.