Proyecto II: MusicBox

Juan José Rodríguez Chaves

Gabriel David Brenes Casasola

Escuela de ingeniería en Computadores, Instituto Tecnológico de Costa Rica

CE-1103: Algoritmos y Estructura de Datos

Leonardo Andres Araya Martinez

13 de enero de 2026

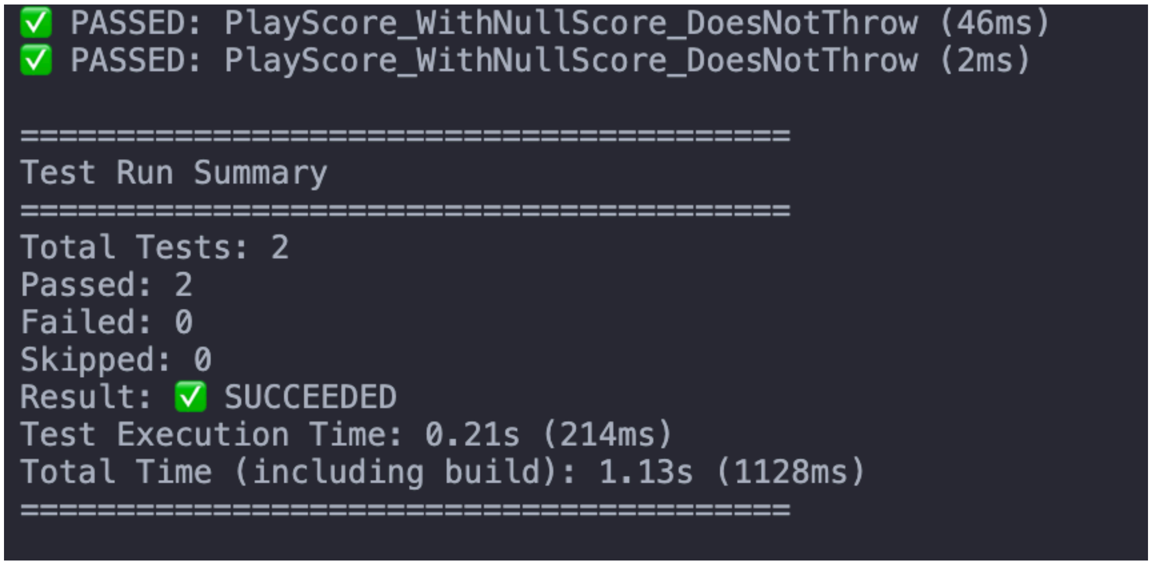
1. Introducción:

El documento en cuestión describe los tests unitarios implementados para validar el correcto funcionamiento del proyecto “MusicBox”, un reproductor de partituras basado en listas doblemente enlazadas. En total se implementaron exactamente 38 tests que cubren todas las funcionalidades principales del sistema.

2. Tests unitarios:

Test 1: AudioEngine.PlayScore_WithNullScore_DoesNotThrow():

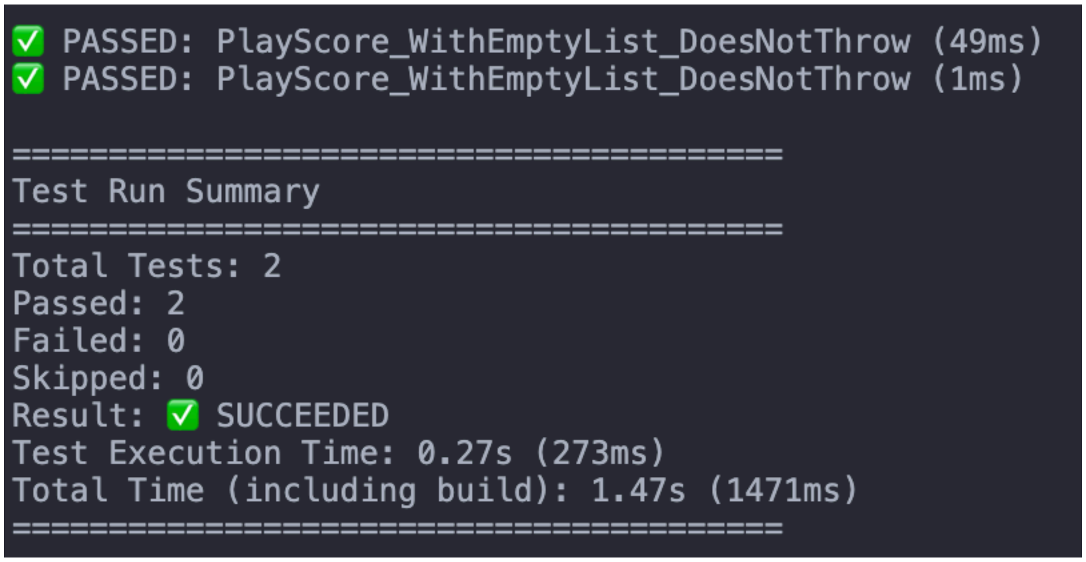
- **Qué hace:** Verifica que el AudioEngine maneje correctamente una partitura nula sin lanzar excepciones.
- **Entrada:** *score = null.*
- **Salida esperada:** No se lanza ninguna excepción y se muestra un mensaje en consola.



```
✅ PASSED: PlayScore_WithNullScore_DoesNotThrow (46ms)
✅ PASSED: PlayScore_WithNullScore_DoesNotThrow (2ms)

========================================
Test Run Summary
========================================
Total Tests: 2
Passed: 2
Failed: 0
Skipped: 0
Result: ✅ SUCCEEDED
Test Execution Time: 0.21s (214ms)
Total Time (including build): 1.13s (1128ms)
========================================
```

Test 2: AudioEngine.PlayScore_WithEmptyList_DoesNotThrow():

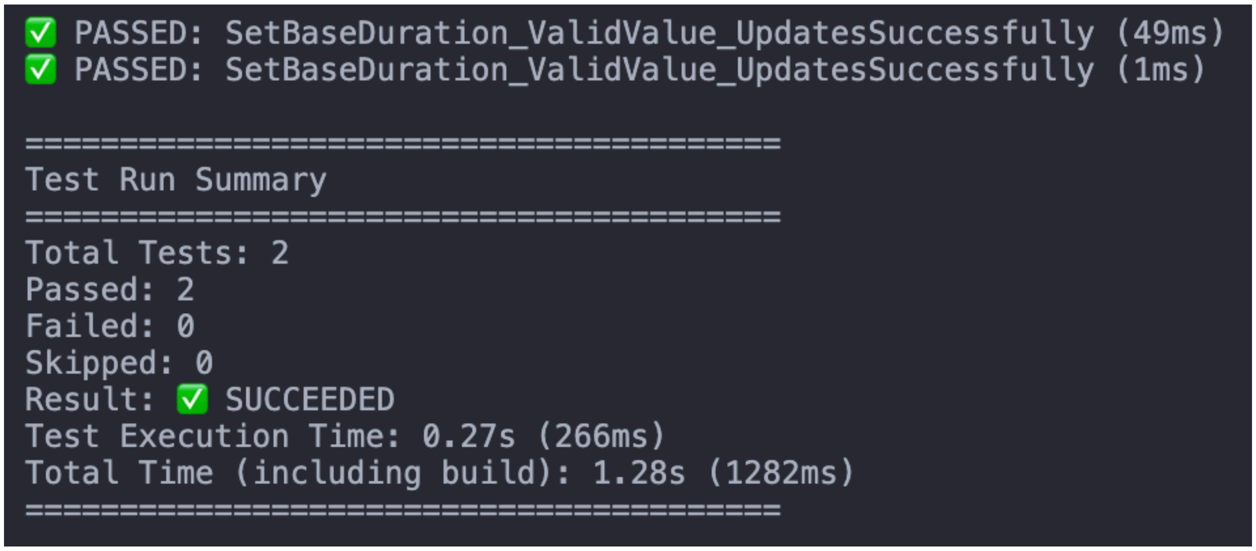
- **Qué hace:** Valida que el sistema reproduzca una lista vacía sin errores.
- **Entrada:** *score = new DoubltLinkedList<Note>( ).*
- **Salida esperada:** No se lanza ninguna excepción.



```
✅ PASSED: PlayScore_WithEmptyList_DoesNotThrow (49ms)
✅ PASSED: PlayScore_WithEmptyList_DoesNotThrow (1ms)

========================================
Test Run Summary
========================================
Total Tests: 2
Passed: 2
Failed: 0
Skipped: 0
Result: ✅ SUCCEEDED
Test Execution Time: 0.27s (273ms)
Total Time (including build): 1.47s (1471ms)
========================================
```

Test 3: AudioEngine.SetBaseDuration_ValidValue_UpdatesSuccesfully():

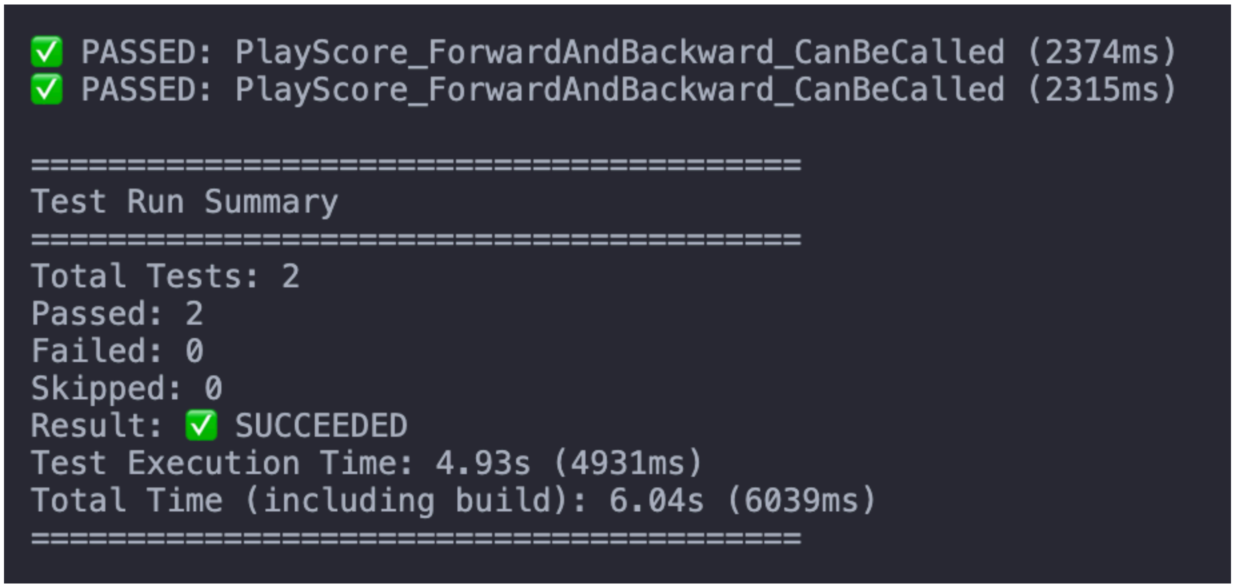
- **Qué hace:** Prueba que se pueda cambiar la duración base de las notas.
- **Entrada:** *newDuration = 0.5.*
- **Salida esperada:** Se actualiza la duración base sin errores.



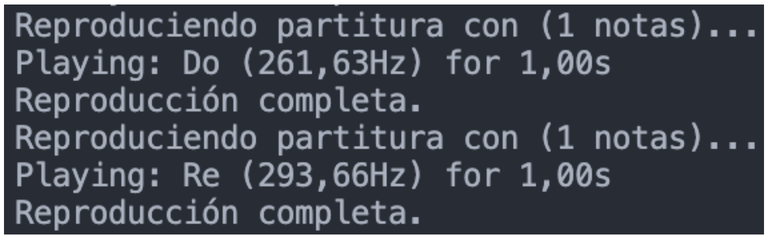
```
✅ PASSED: SetBaseDuration_ValidValue_UpdatesSuccessfully (49ms)
✅ PASSED: SetBaseDuration_ValidValue_UpdatesSuccessfully (1ms)

========================================
Test Run Summary
========================================
Total Tests: 2
Passed: 2
Failed: 0
Skipped: 0
Result: ✅ SUCCEEDED
Test Execution Time: 0.27s (266ms)
Total Time (including build): 1.28s (1282ms)
========================================
```

Test 4: AudioEngine.PlayScore_ForwardAndBackward_CanBeCalled():

- **Qué hace:** Verifica que se puedan reproducir partituras en ambas direcciones
- **Entrada:** *Dos partituras simples (Do y Re) reproduciendo hacia adelante y hacia atrás respectivamente.*
- **Salida esperada:** No se lanza ninguna excepciones en ninguna dirección.



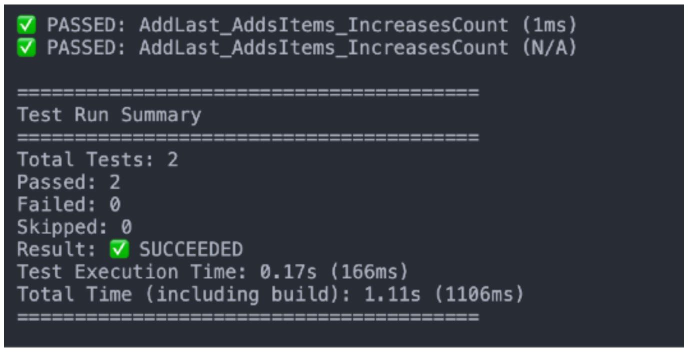
```
✅ PASSED: PlayScore_ForwardAndBackward_CanBeCalled (2374ms)
✅ PASSED: PlayScore_ForwardAndBackward_CanBeCalled (2315ms)

========================================
Test Run Summary
========================================
Total Tests: 2
Passed: 2
Failed: 0
Skipped: 0
Result: ✅ SUCCEEDED
Test Execution Time: 4.93s (4931ms)
Total Time (including build): 6.04s (6039ms)
========================================
```



```
Reproduciendo partitura con (1 notas)...
Playing: Do (261,63Hz) for 1,00s
Reproducción completa.
Reproduciendo partitura con (1 notas)...
Playing: Re (293,66Hz) for 1,00s
Reproducción completa.
```

Test 5: DoublyLinkedList.AddLast_AddsItems_IncreaseCount():

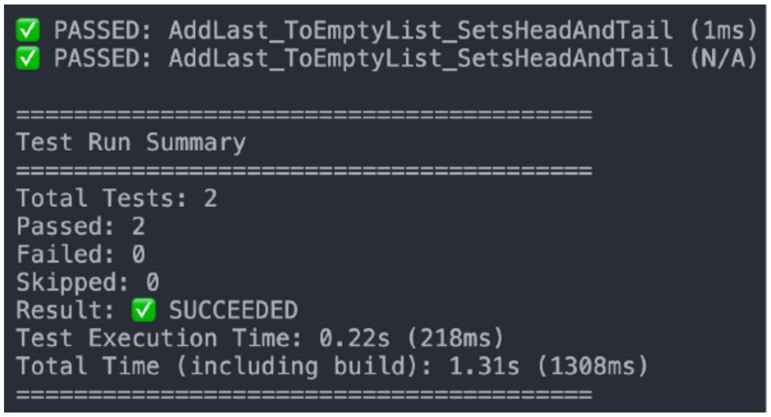
- **Qué hace:** Prueba que al agregar elementos, el contador de la lista aumente.
- **Entrada:** *list.AddLast(1); list.AddLast(2); list.AddLast(3);.*
- **Salida esperada:** list.Count == 3.



```
✅ PASSED: AddLast_AddsItems_IncreasesCount (1ms)
✅ PASSED: AddLast_AddsItems_IncreasesCount (N/A)

========================================
Test Run Summary
========================================
Total Tests: 2
Passed: 2
Failed: 0
Skipped: 0
Result: ✅ SUCCEEDED
Test Execution Time: 0.17s (166ms)
Total Time (including build): 1.11s (1106ms)
========================================
```

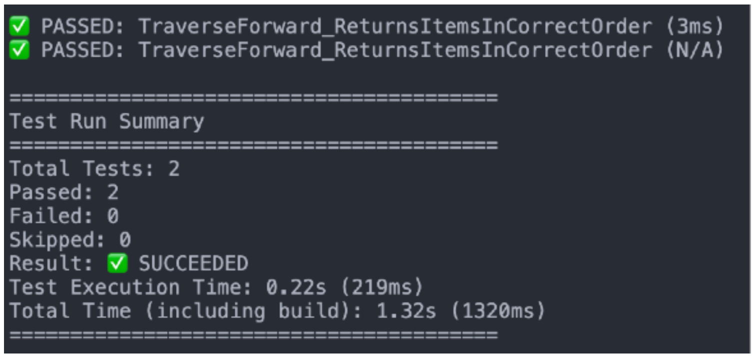
Test 6: DoublyLinkedList.ToEmptyList_SetsHeadAndTail():

- **Qué hace:** Verifica que al agregar un elemento a una lista vacía, se configuren cabeza y cola correctamente.
- **Entrada:** *list.AddLast("Test") en lista vacía.*
- **Salida esperada: Head** y **Tail** apuntan al mismo nodo con datos "Test".



```
✅ PASSED: AddLast_ToEmptyList_SetsHeadAndTail (1ms)
✅ PASSED: AddLast_ToEmptyList_SetsHeadAndTail (N/A)

========================================
Test Run Summary
========================================
Total Tests: 2
Passed: 2
Failed: 0
Skipped: 0
Result: ✅ SUCCEEDED
Test Execution Time: 0.22s (218ms)
Total Time (including build): 1.31s (1308ms)
========================================
```

Test 7: DoublyLinkedList.TraverseForward_ReturnsItemsInCorrectOrder():

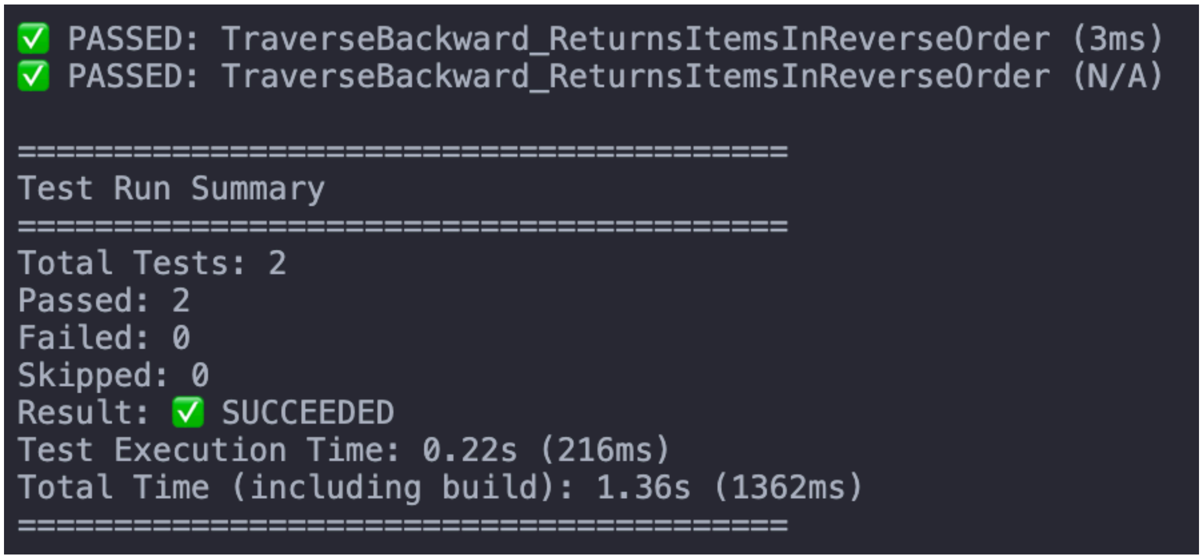


```
✅ PASSED: TraverseForward_ReturnsItemsInCorrectOrder (3ms)
✅ PASSED: TraverseForward_ReturnsItemsInCorrectOrder (N/A)

========================================
Test Run Summary
========================================
Total Tests: 2
Passed: 2
Failed: 0
Skipped: 0
Result: ✅ SUCCEEDED
Test Execution Time: 0.22s (219ms)
Total Time (including build): 1.32s (1320ms)
========================================
```

- **Qué hace:** Prueba que el recorrido hacia adelanteretorne elementos en orden de inserción.
- **Entrada:** *Lista con [10, 20, 30].*
- **Salida esperada:** Secuencia [10, 20, 30].

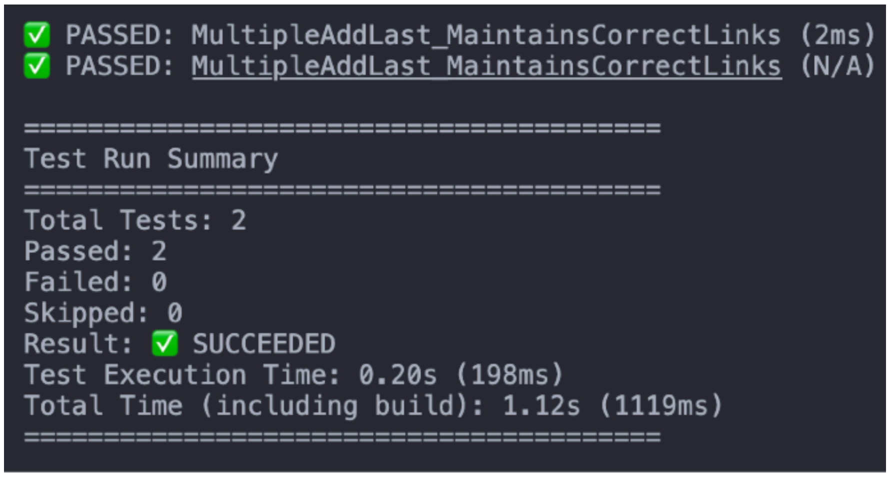
Test 8: DoublyLinkedList.TraverseBackward_ReturnsItemsInReverseOrder():

- **Qué hace:** Valida que el recorrido hacia atrás retorne elementos en orden inverso.
- **Entrada:** *Lista con [10, 20, 30].*
- **Salida esperada:** Secuencia [30, 20, 10].



```
✅ PASSED: TraverseBackward_ReturnsItemsInReverseOrder (3ms)
✅ PASSED: TraverseBackward_ReturnsItemsInReverseOrder (N/A)

========================================
Test Run Summary
========================================
Total Tests: 2
Passed: 2
Failed: 0
Skipped: 0
Result: ✅ SUCCEEDED
Test Execution Time: 0.22s (216ms)
Total Time (including build): 1.36s (1362ms)
========================================
```

Test 9: DoublyLinkedList.MultipleAddLast_MantainsCorrectLinks():



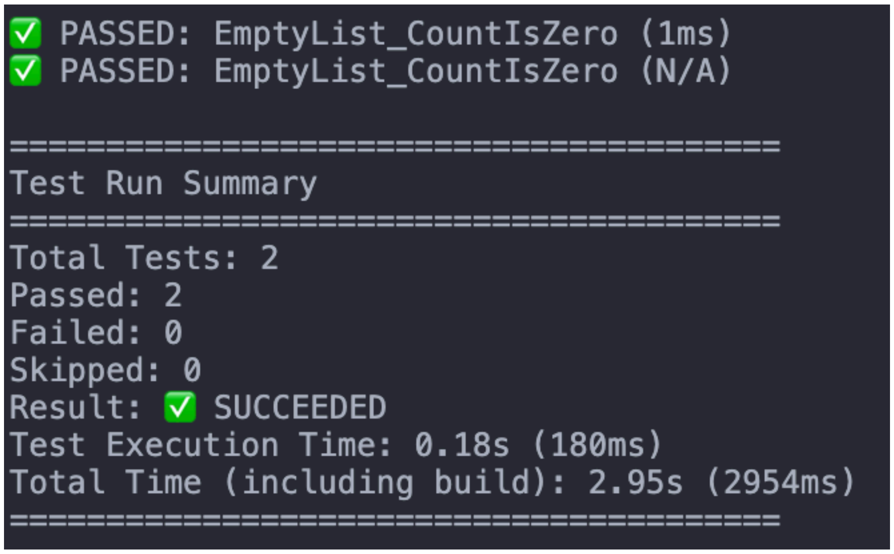
```
✅ PASSED: MultipleAddLast_MaintainsCorrectLinks (2ms)
✅ PASSED: MultipleAddLast_MaintainsCorrectLinks (N/A)

========================================
Test Run Summary
========================================
Total Tests: 2
Passed: 2
Failed: 0
Skipped: 0
Result: ✅ SUCCEEDED
Test Execution Time: 0.20s (198ms)
Total Time (including build): 1.12s (1119ms)
========================================
```

- **Qué hace:** Verifica que los enlaces entre nodos se mantengan correctamente.
- **Entrada:** *Lista con tres elementos [10, 20, 30].*
- **Salida esperada:** Enlaces **Previous** y **Next** correctos entre todos los nodos.

Test 10: DoublyLinkedList.EmptyList_CountIsZero():

- **Qué hace:** Prueba que una lista vacía tenga contador cero.
- **Entrada:** *Lista vacía.*

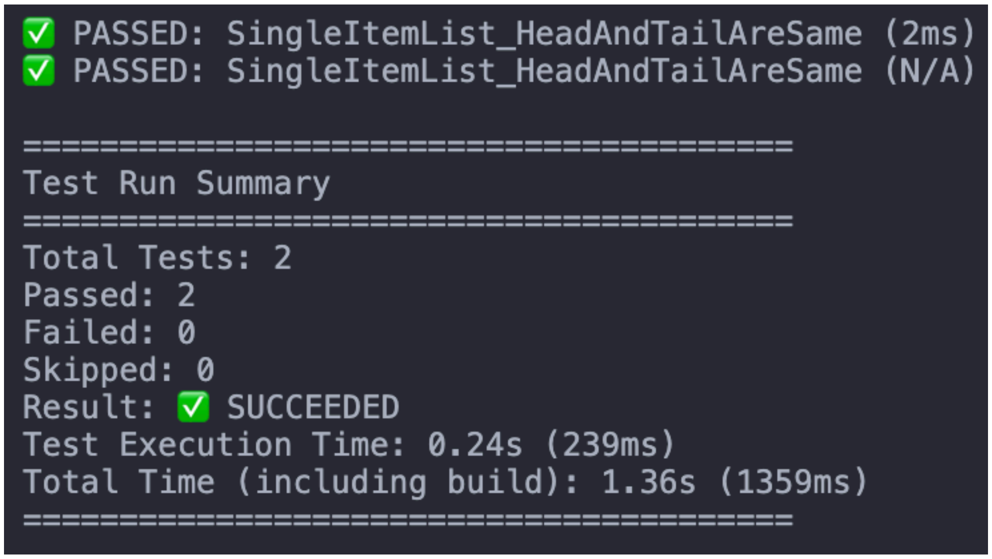
- **Salida esperada:** Count == 0, Head == null, Tail == null.



```
✅ PASSED: EmptyList_CountIsZero (1ms)
✅ PASSED: EmptyList_CountIsZero (N/A)

========================================
Test Run Summary
========================================
Total Tests: 2
Passed: 2
Failed: 0
Skipped: 0
Result: ✅ SUCCEEDED
Test Execution Time: 0.18s (180ms)
Total Time (including build): 2.95s (2954ms)
========================================
```

Test 11: DoublyLinkedList.SingleItemList_HeadAndTailAreSame():

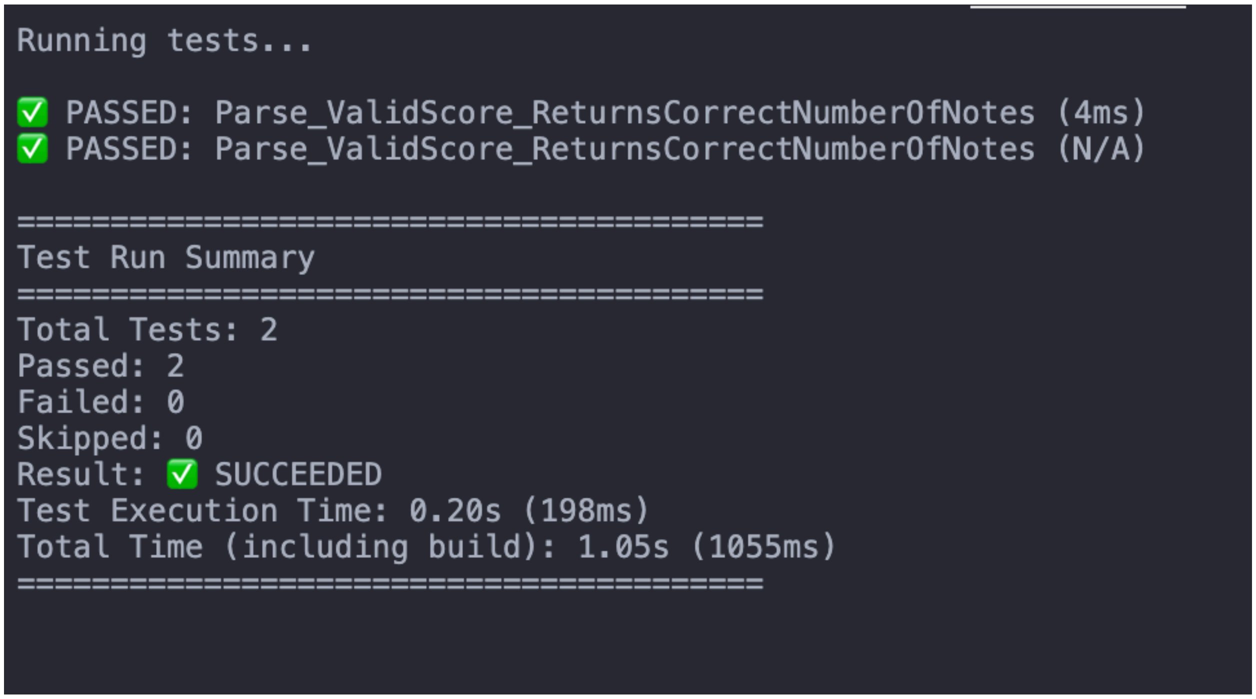
- **Qué hace:** Verifica que en una lista con un solo elemento, cabeza y cola sean iguales
- **Entrada:** *list.AddLast(42).*
- **Salida esperada:** Head == Tail y contienen el valor 42.



```
✅ PASSED: SingleItemList_HeadAndTailAreSame (2ms)
✅ PASSED: SingleItemList_HeadAndTailAreSame (N/A)

========================================
Test Run Summary
========================================
Total Tests: 2
Passed: 2
Failed: 0
Skipped: 0
Result: ✅ SUCCEEDED
Test Execution Time: 0.24s (239ms)
Total Time (including build): 1.36s (1359ms)
========================================
```

Test 12: ScoreParserTests.Parse_ValidScore_ReturnsCorrectNumberOfNotes():

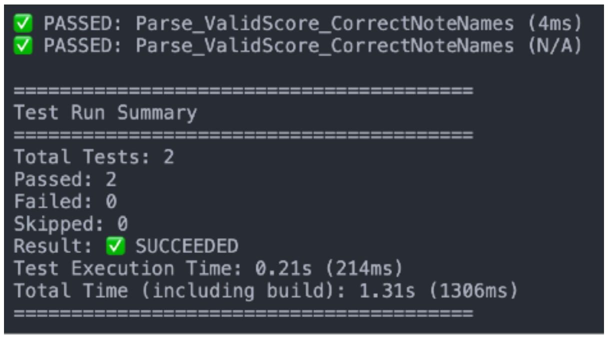
- **Qué hace:** Verifica que el parser procese el número correcto de notas.
- **Entrada:** *"(Do, negra), (Re, blanca), (Mi, corchea)"*
- **Salida esperada:** result.Count == 3



```
Running tests...

✅ PASSED: Parse_ValidScore_ReturnsCorrectNumberOfNotes (4ms)
✅ PASSED: Parse_ValidScore_ReturnsCorrectNumberOfNotes (N/A)

========================================
Test Run Summary
========================================
Total Tests: 2
Passed: 2
Failed: 0
Skipped: 0
Result: ✅ SUCCEEDED
Test Execution Time: 0.20s (198ms)
Total Time (including build): 1.05s (1055ms)
========================================
```

Test 13: ScoreParserTests.Parse_ValidScore_CorrectNoteNames():

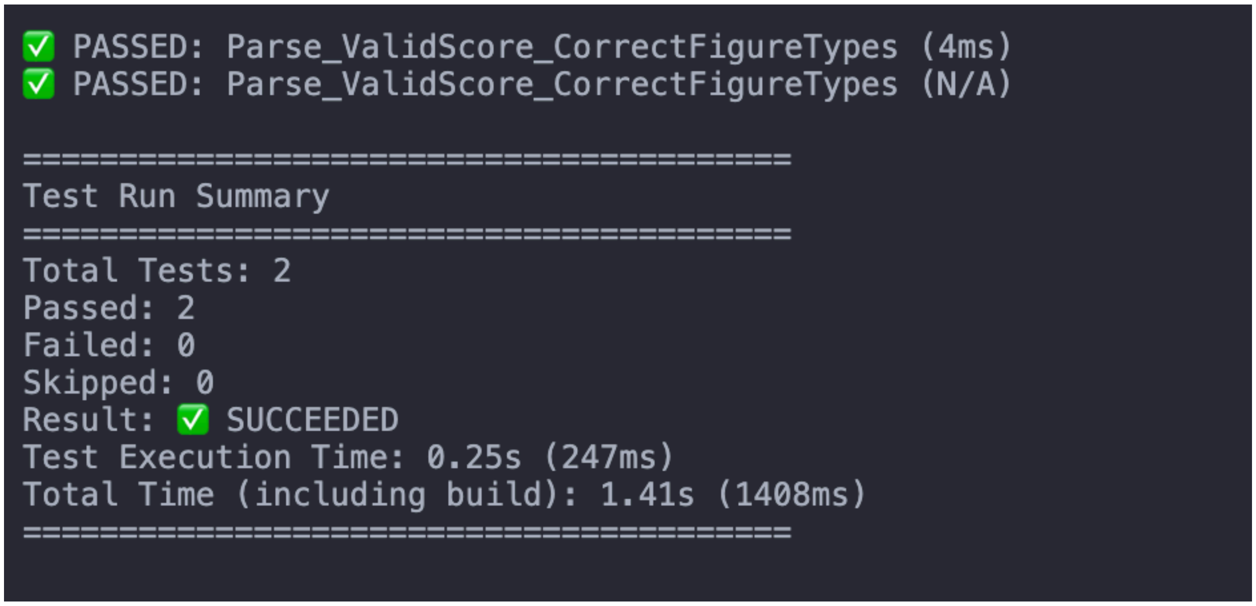
- **Qué hace:** Prueba que los nombres de notas se parseen correctamente.
- **Entrada:** *"(Do, negra), (Re, blanca), (Mi, corchea)".*
- **Salida esperada:** Notas en orden: Do, Re, Mi.



```
✅ PASSED: Parse_ValidScore_CorrectNoteNames (4ms)
✅ PASSED: Parse_ValidScore_CorrectNoteNames (N/A)

========================================
Test Run Summary
========================================
Total Tests: 2
Passed: 2
Failed: 0
Skipped: 0
Result: ✅ SUCCEEDED
Test Execution Time: 0.21s (214ms)
Total Time (including build): 1.31s (1306ms)
========================================
```

Test 14: ScoreParserTests.Parse_ValidScore_CorrectFigureTypes():

- **Qué hace:** Valida que las figuras musicales se parseen correctamente.
- **Entrada:** *"(Do, negra), (Re, blanca), (Mi, corchea)".*
- **Salida esperada:** Figuras en orden: **Negra, Blanca, Corchea.**



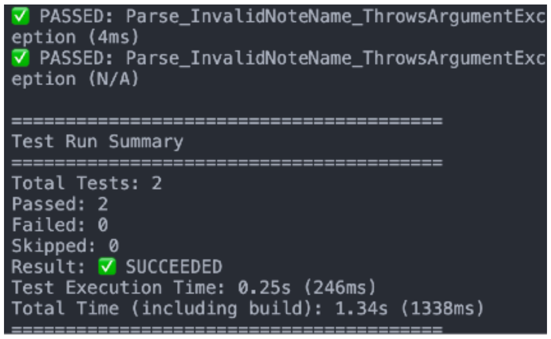
```
✅ PASSED: Parse_ValidScore_CorrectFigureTypes (4ms)
✅ PASSED: Parse_ValidScore_CorrectFigureTypes (N/A)

========================================
Test Run Summary
========================================
Total Tests: 2
Passed: 2
Failed: 0
Skipped: 0
Result: ✅ SUCCEEDED
Test Execution Time: 0.25s (247ms)
Total Time (including build): 1.41s (1408ms)
========================================
```

Test 15:

ScoreParserTests.Parse_InvalidNoteName_ThrowsArgumentException()

:

- **Qué hace:** Prueba que se lance excepción con nombres de nota inválidos.
- **Entrada:** *("Do, negra), (Xi, blanca)", en este caso "Xi" no es válido.*
- **Salida esperada:** Se lanza **ArgumentException**.



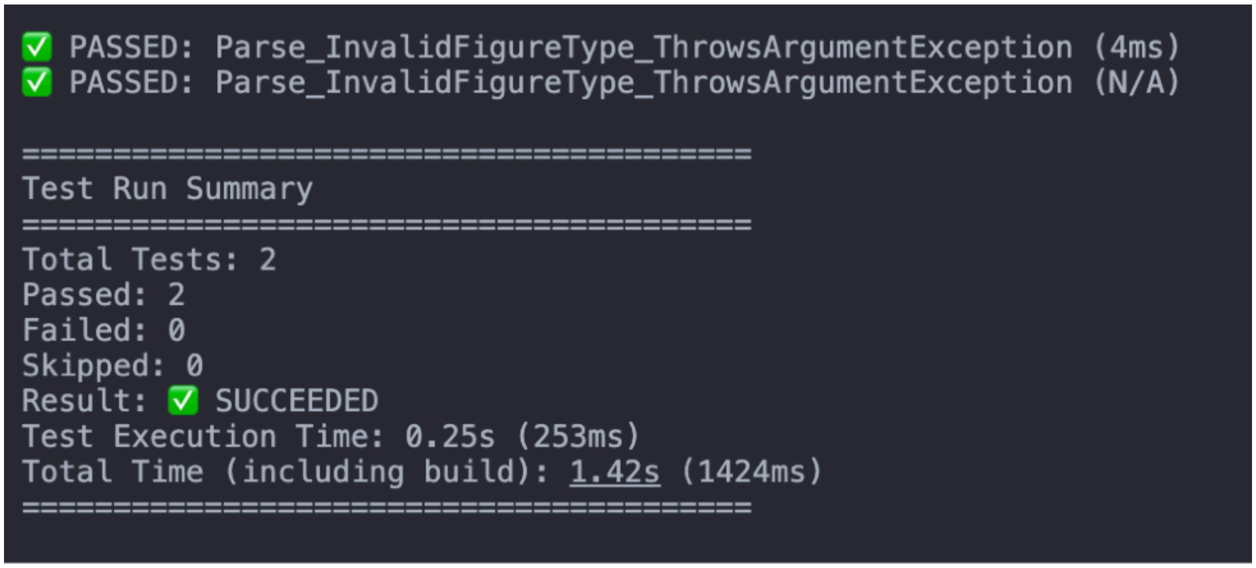
```
✅ PASSED: Parse_InvalidNoteName_ThrowsArgumentExc
eption (4ms)
✅ PASSED: Parse_InvalidNoteName_ThrowsArgumentExc
eption (N/A)

========================================
Test Run Summary
========================================
Total Tests: 2
Passed: 2
Failed: 0
Skipped: 0
Result: ✅ SUCCEEDED
Test Execution Time: 0.25s (246ms)
Total Time (including build): 1.34s (1338ms)
========================================
```

Test 16:

ScoreParserTests.Parse_InvalidFigureType_ThrowsArgumentException()

:

- **Qué hace:** Valida que se lance excepción con figuras inválidas.
- **Entrada:** *"(Do, negra), (Re, invalida)", en este caso "invalido" no es una figura válida.*
- **Salida esperada:** Se lanza **ArgumentException.**



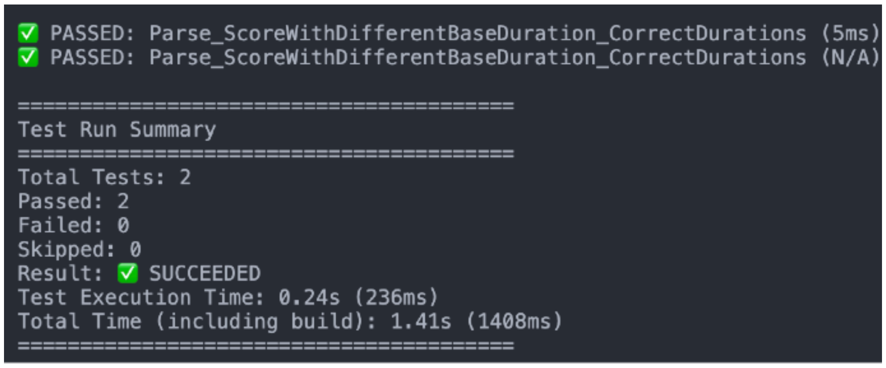
```
✅ PASSED: Parse_InvalidFigureType_ThrowsArgumentException (4ms)
✅ PASSED: Parse_InvalidFigureType_ThrowsArgumentException (N/A)

========================================
Test Run Summary
========================================
Total Tests: 2
Passed: 2
Failed: 0
Skipped: 0
Result: ✅ SUCCEEDED
Test Execution Time: 0.25s (253ms)
Total Time (including build): 1.42s (1424ms)
========================================
```

Test

17:ScoreParserTests.Parse_ScoreWithDifferentBaseDuration_CorrectDurations():

- **Qué hace:** Prueba que las duraciones se calculen correctamente con base personalizada.
- **Entrada:** (*baseDuration = 2.0), partitura con tres notas.*

- **Salida esperada:** Duraciones de **2.0s, 4.0s y 1.0s** respectivamente.

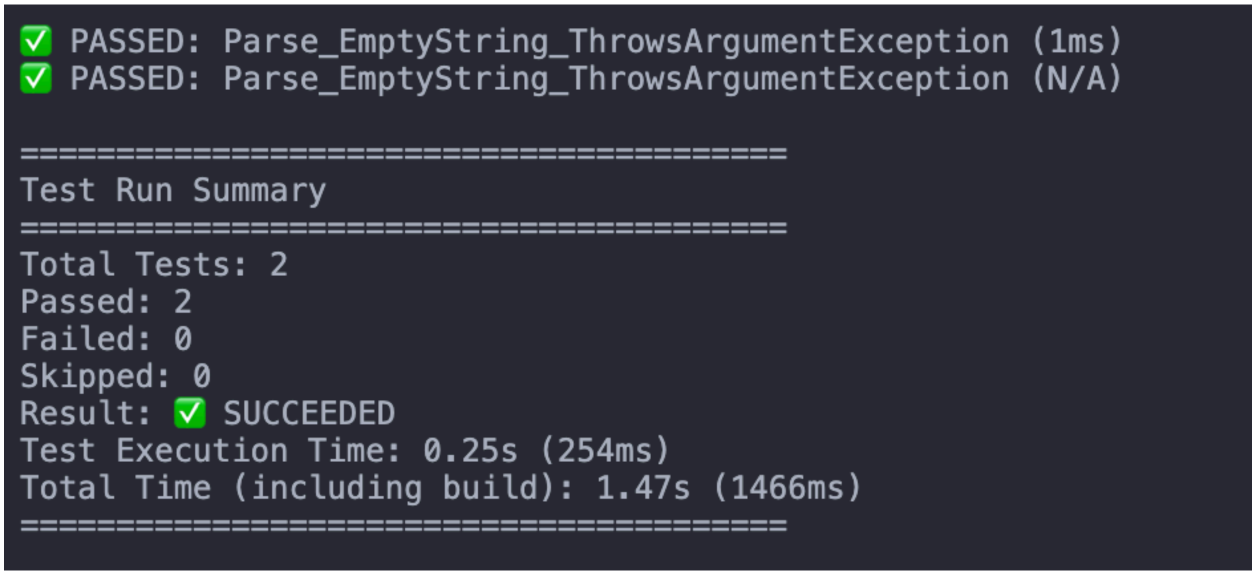


```
✅ PASSED: Parse_ScoreWithDifferentBaseDuration_CorrectDurations (5ms)
✅ PASSED: Parse_ScoreWithDifferentBaseDuration_CorrectDurations (N/A)

========================================
Test Run Summary
========================================
Total Tests: 2
Passed: 2
Failed: 0
Skipped: 0
Result: ✅ SUCCEEDED
Test Execution Time: 0.24s (236ms)
Total Time (including build): 1.41s (1408ms)
========================================
```

Test 18:

ScoreParserTests.Parse_EmptyString_ThrowsArgumentException():

- **Qué hace:** Verifica que string vacío lance excepción.
- **Entrada:** *“”.*
- **Salida esperada:** Se lanza **FormatException.**

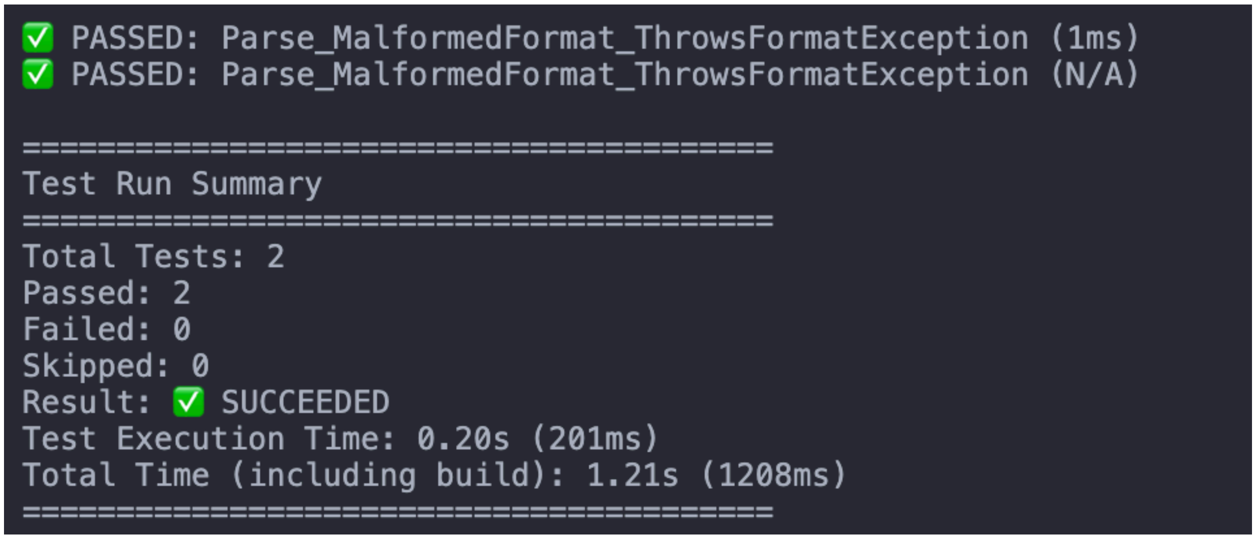


```
✅ PASSED: Parse_EmptyString_ThrowsArgumentException (1ms)
✅ PASSED: Parse_EmptyString_ThrowsArgumentException (N/A)

========================================
Test Run Summary
========================================
Total Tests: 2
Passed: 2
Failed: 0
Skipped: 0
Result: ✅ SUCCEEDED
Test Execution Time: 0.25s (254ms)
Total Time (including build): 1.47s (1466ms)
========================================
```

Test 19:

ScoreParserTests.Parse_MalformedFormat_ThrowsFormatException():

- **Qué hace:** Prueba que un formato mal escrito lance excepción.
- **Entrada:** *("(Do negra), (Re, blanca)"), falta una coma.*
- **Salida esperada:** Se lanza **FormatException.**

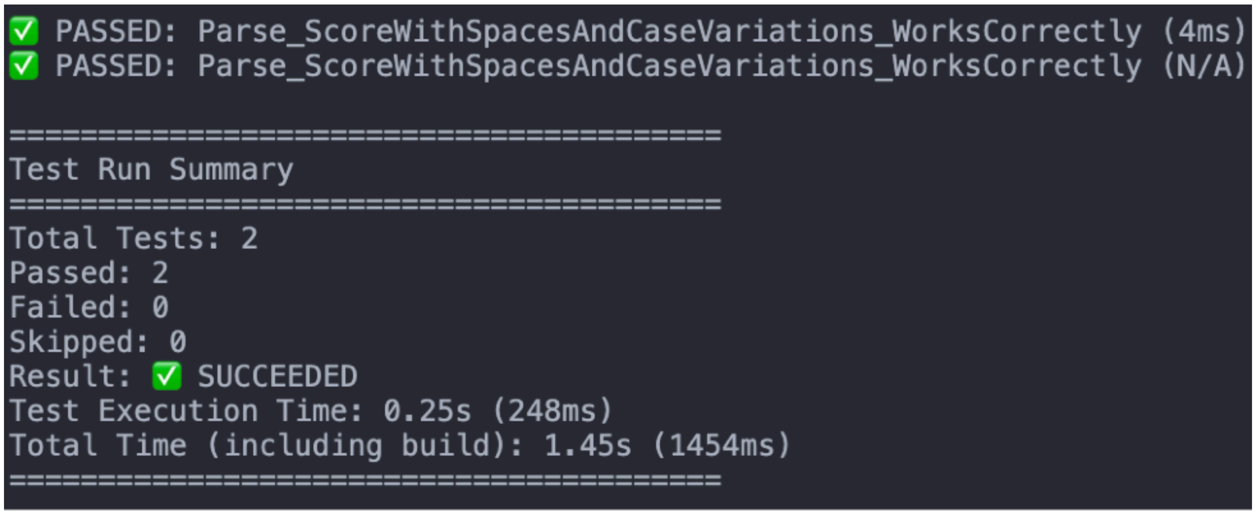


```
✅ PASSED: Parse_MalformedFormat_ThrowsFormatException (1ms)
✅ PASSED: Parse_MalformedFormat_ThrowsFormatException (N/A)

========================================
Test Run Summary
========================================
Total Tests: 2
Passed: 2
Failed: 0
Skipped: 0
Result: ✅ SUCCEEDED
Test Execution Time: 0.20s (201ms)
Total Time (including build): 1.21s (1208ms)
========================================
```

Test 20:

ScoreParserTests.Parse_ScoreWithSpacesAndCaseVariations_WorkCorrectly():

- **Qué hace:** Verifica que el parser maneje variaciones de espacio y mayúsculas.
- **Entrada:** *"(Do, NEGRA), (re, blanca), (Mi, CORCHEA)".*
- **Salida esperada:** 3 notas pareadas correctamente.

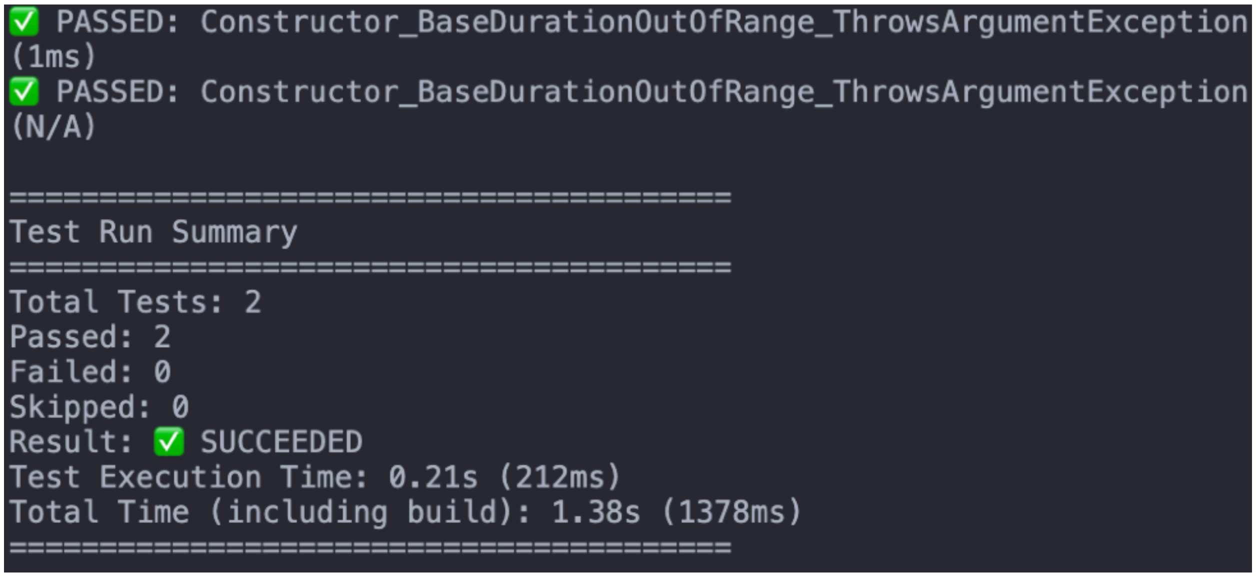


```
✅ PASSED: Parse_ScoreWithSpacesAndCaseVariations_WorksCorrectly (4ms)
✅ PASSED: Parse_ScoreWithSpacesAndCaseVariations_WorksCorrectly (N/A)

========================================
Test Run Summary
========================================
Total Tests: 2
Passed: 2
Failed: 0
Skipped: 0
Result: ✅ SUCCEEDED
Test Execution Time: 0.25s (248ms)
Total Time (including build): 1.45s (1454ms)
========================================
```

Test 21:

Score.ParserTests.Constructor_BaseDurationOutofRange_ThrowsArgumentException():

- **Qué hace:** Verifica que duraciones base fuera del rango lancen excepción.
- **Entrada:** (*new ScoreParser(0.05)) y (new ScoreParser(6.0)).*
- **Salida esperada:** Se lanza **ArgumentException** en ambos casos.

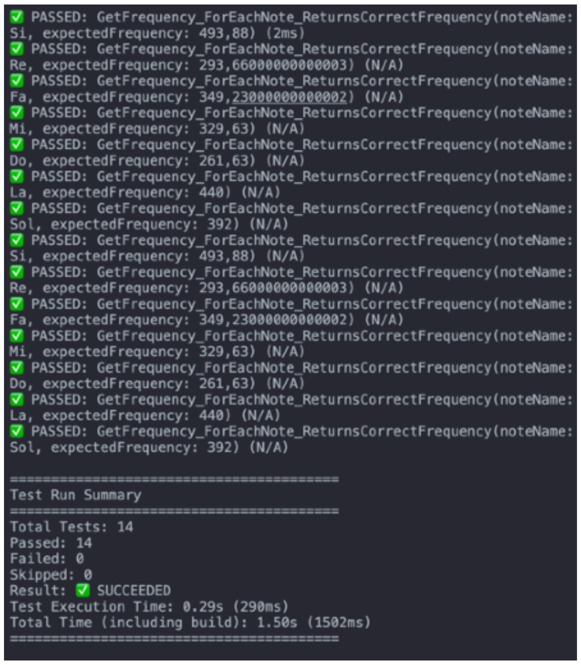


```
✅ PASSED: Constructor_BaseDurationOutOfRange_ThrowsArgumentException
(1ms)
✅ PASSED: Constructor_BaseDurationOutOfRange_ThrowsArgumentException
(N/A)

========================================
Test Run Summary
========================================
Total Tests: 2
Passed: 2
Failed: 0
Skipped: 0
Result: ✅ SUCCEEDED
Test Execution Time: 0.21s (212ms)
Total Time (including build): 1.38s (1378ms)
========================================
```

Test 22-28:

Note.GetFrequency_ForEachNote_ReturnsCorrectFrequency():

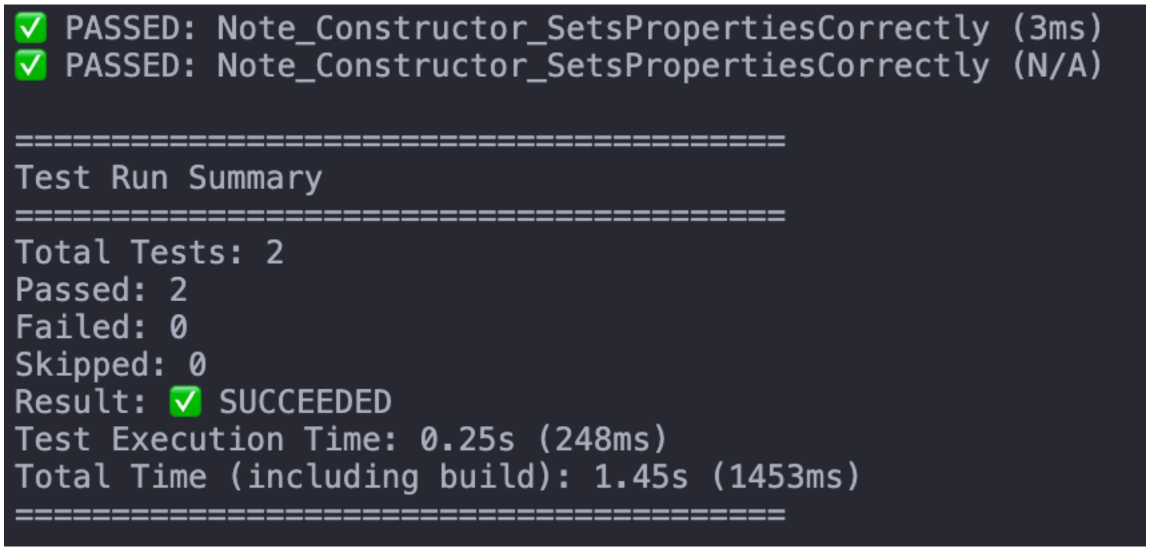
- **Qué hace:** Valida que cada nota musical tenga su frecuencia correcta.
- **Entradas y salidas esperadas:**
    1. Do -> 261.63 Hz.
    2. Re -> 293.66 Hz.
    3. Mi -> 329.63 Hz.
    4. Fa -> 349.23 Hz.
    5. Sol -> 392.00 Hz.
    6. La -> 440.00 Hz.
    7. Si -> 493.88 Hz.



```
✅ PASSED: GetFrequency_ForEachNote_ReturnsCorrectFrequency(noteName:
Si, expectedFrequency: 493,88) (2ms)
✅ PASSED: GetFrequency_ForEachNote_ReturnsCorrectFrequency(noteName:
Re, expectedFrequency: 293,66000000000003) (N/A)
✅ PASSED: GetFrequency_ForEachNote_ReturnsCorrectFrequency(noteName:
Fa, expectedFrequency: 349,23000000000002) (N/A)
✅ PASSED: GetFrequency_ForEachNote_ReturnsCorrectFrequency(noteName:
Mi, expectedFrequency: 329,63) (N/A)
✅ PASSED: GetFrequency_ForEachNote_ReturnsCorrectFrequency(noteName:
Do, expectedFrequency: 261,63) (N/A)
✅ PASSED: GetFrequency_ForEachNote_ReturnsCorrectFrequency(noteName:
La, expectedFrequency: 440) (N/A)
✅ PASSED: GetFrequency_ForEachNote_ReturnsCorrectFrequency(noteName:
Sol, expectedFrequency: 392) (N/A)
✅ PASSED: GetFrequency_ForEachNote_ReturnsCorrectFrequency(noteName:
Si, expectedFrequency: 493,88) (N/A)
✅ PASSED: GetFrequency_ForEachNote_ReturnsCorrectFrequency(noteName:
Re, expectedFrequency: 293,66000000000003) (N/A)
✅ PASSED: GetFrequency_ForEachNote_ReturnsCorrectFrequency(noteName:
Fa, expectedFrequency: 349,23000000000002) (N/A)
✅ PASSED: GetFrequency_ForEachNote_ReturnsCorrectFrequency(noteName:
Mi, expectedFrequency: 329,63) (N/A)
✅ PASSED: GetFrequency_ForEachNote_ReturnsCorrectFrequency(noteName:
Do, expectedFrequency: 261,63) (N/A)
✅ PASSED: GetFrequency_ForEachNote_ReturnsCorrectFrequency(noteName:
La, expectedFrequency: 440) (N/A)
✅ PASSED: GetFrequency_ForEachNote_ReturnsCorrectFrequency(noteName:
Sol, expectedFrequency: 392) (N/A)

========================================
Test Run Summary
========================================
Total Tests: 14
Passed: 14
Failed: 0
Skipped: 0
Result: ✅ SUCCEEDED
Test Execution Time: 0.29s (290ms)
Total Time (including build): 1.50s (1502ms)
========================================
```

Test 29: Note.Note_Constructor_SetsPropertiesCorrectly():

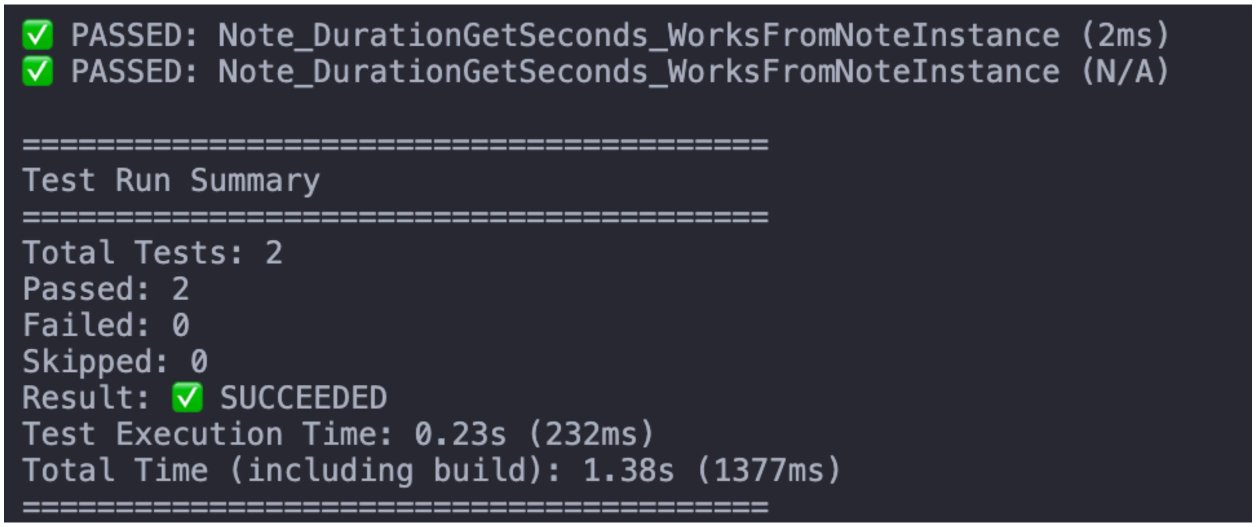
- **Qué hace:** Prueba que la construcción de notas asigne propiedades correctamente.
- **Entrada:** *Nota Do con frecuencia 261.63 y duración de negra (1.0s).*
- **Salida esperada:** Todas las propiedades asignadas correctamente.



```
✅ PASSED: Note_Constructor_SetsPropertiesCorrectly (3ms)
✅ PASSED: Note_Constructor_SetsPropertiesCorrectly (N/A)

========================================
Test Run Summary
========================================
Total Tests: 2
Passed: 2
Failed: 0
Skipped: 0
Result: ✅ SUCCEEDED
Test Execution Time: 0.25s (248ms)
Total Time (including build): 1.45s (1453ms)
========================================
```

Test 30: Note.Note_DurationGetSeconds_WorksFromNoteInstance():

- **Qué hace:** Verifica que se pueda obtener duración desde instancia de nota.
- **Entrada:** *Nota Do con figura Blanca y base 0.5s.*
- **Salida esperada:** Duración de 1.0s (2 x 0.5).

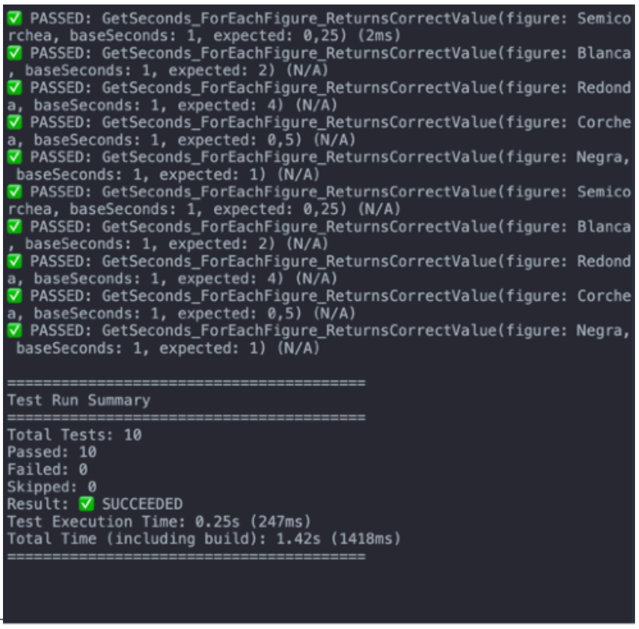


```
✅ PASSED: Note_DurationGetSeconds_WorksFromNoteInstance (2ms)
✅ PASSED: Note_DurationGetSeconds_WorksFromNoteInstance (N/A)

========================================
Test Run Summary
========================================
Total Tests: 2
Passed: 2
Failed: 0
Skipped: 0
Result: ✅ SUCCEEDED
Test Execution Time: 0.23s (232ms)
Total Time (including build): 1.38s (1377ms)
========================================
```

Test 31-35:

Duration.GetSeconds_ForEachFigure_ReturnsCorrectValue():

- **Qué hace:** Verifica cálculo de duración para cada figura musical.
- **Entrada y salidas esperadas (con base 1.0s):**
  1. Redonda -> 4.0s.
  2. Blanca -> 2.0s.
  3. Negra -> 1.0s.
  4. Corchea -> 0.5s.
  5. Semicorchea -> 0.25s.



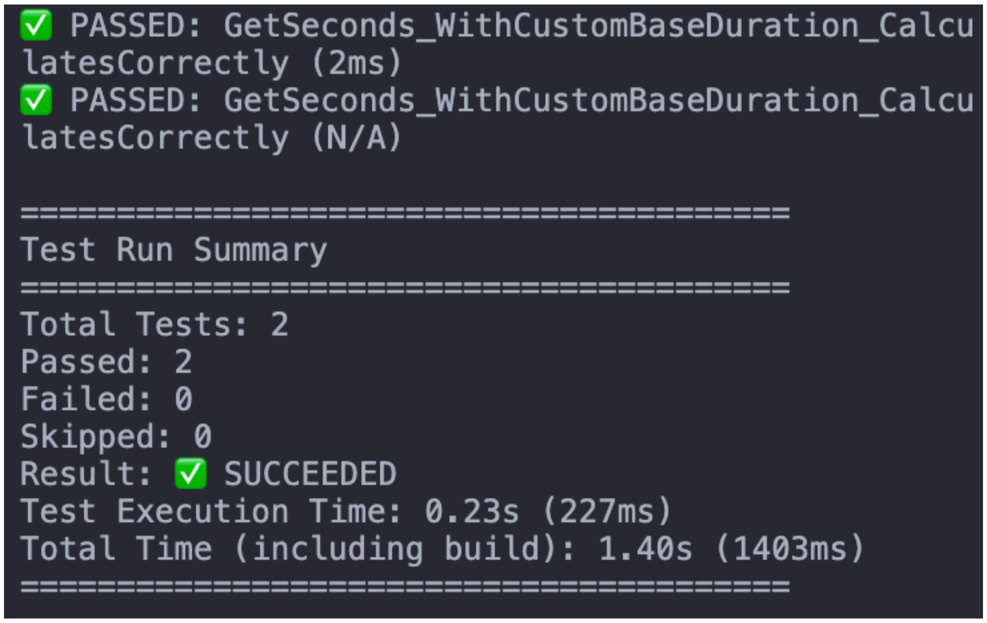
```
✅ PASSED: GetSeconds_ForEachFigure_ReturnsCorrectValue(figure: Semico
rchea, baseSeconds: 1, expected: 0,25) (2ms)
✅ PASSED: GetSeconds_ForEachFigure_ReturnsCorrectValue(figure: Blanca
, baseSeconds: 1, expected: 2) (N/A)
✅ PASSED: GetSeconds_ForEachFigure_ReturnsCorrectValue(figure: Redond
a, baseSeconds: 1, expected: 4) (N/A)
✅ PASSED: GetSeconds_ForEachFigure_ReturnsCorrectValue(figure: Corche
a, baseSeconds: 1, expected: 0,5) (N/A)
✅ PASSED: GetSeconds_ForEachFigure_ReturnsCorrectValue(figure: Negra,
 baseSeconds: 1, expected: 1) (N/A)
✅ PASSED: GetSeconds_ForEachFigure_ReturnsCorrectValue(figure: Semico
rchea, baseSeconds: 1, expected: 0,25) (N/A)
✅ PASSED: GetSeconds_ForEachFigure_ReturnsCorrectValue(figure: Blanca
, baseSeconds: 1, expected: 2) (N/A)
✅ PASSED: GetSeconds_ForEachFigure_ReturnsCorrectValue(figure: Redond
a, baseSeconds: 1, expected: 4) (N/A)
✅ PASSED: GetSeconds_ForEachFigure_ReturnsCorrectValue(figure: Corche
a, baseSeconds: 1, expected: 0,5) (N/A)
✅ PASSED: GetSeconds_ForEachFigure_ReturnsCorrectValue(figure: Negra,
 baseSeconds: 1, expected: 1) (N/A)

========================================
Test Run Summary
========================================
Total Tests: 10
Passed: 10
Failed: 0
Skipped: 0
Result: ✅ SUCCEEDED
Test Execution Time: 0.25s (247ms)
Total Time (including build): 1.42s (1418ms)
========================================
```

Test 36:

Duration.GetSeconds_WithCustomBaseDuration_CalculatesCorrectly():

- **Qué hace:** Prueba cálculo con duración base personalizada.
- **Entrada:** *Figura “Negra” con base 0.5s.*

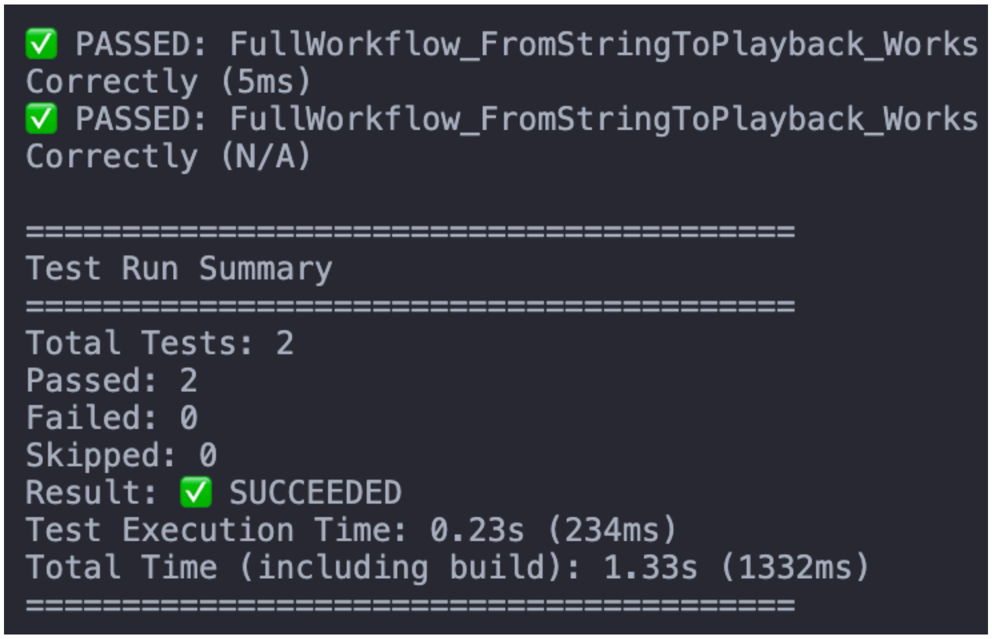
- **Salida esperada:** Duración de 0.5s.



```
✅ PASSED: GetSeconds_WithCustomBaseDuration_Calcu
latesCorrectly (2ms)
✅ PASSED: GetSeconds_WithCustomBaseDuration_Calcu
latesCorrectly (N/A)

========================================
Test Run Summary
========================================
Total Tests: 2
Passed: 2
Failed: 0
Skipped: 0
Result: ✅ SUCCEEDED
Test Execution Time: 0.23s (227ms)
Total Time (including build): 1.40s (1403ms)
========================================
```

Test 37: FullWorkflow_FromStringToPlayback_WorksCorrectly():

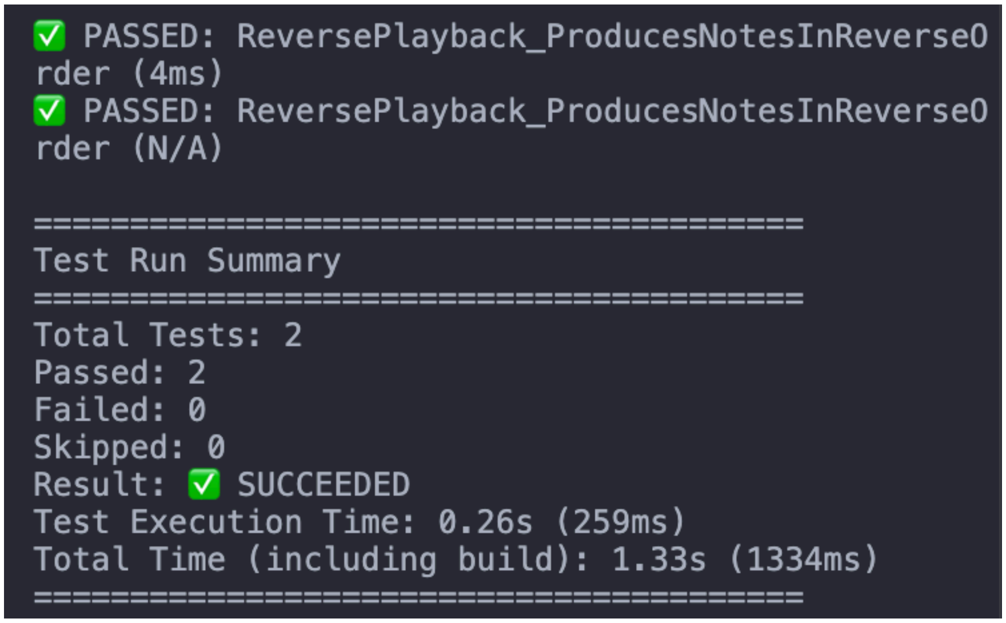
- **Qué hace:** Prueba flujo completo desde storing hasta lista de notas.
- **Entrada:** *"(Do, negra), (Re, blanca), (Mi, corchea)".*
- **Salida esperada:** Lista con 3 notas, todas propiedades correctas (nombre, figura, duración, frecuencia).



```
✅ PASSED: FullWorkflow_FromStringToPlayback_Works
Correctly (5ms)
✅ PASSED: FullWorkflow_FromStringToPlayback_Works
Correctly (N/A)

========================================
Test Run Summary
========================================
Total Tests: 2
Passed: 2
Failed: 0
Skipped: 0
Result: ✅ SUCCEEDED
Test Execution Time: 0.23s (234ms)
Total Time (including build): 1.33s (1332ms)
========================================
```

Test 38: ReversePlayBack_ProduciesNotesInReverseOrder():

- **Qué hace:** Valida que reproducción inversa funcione correctamente.
- **Entrada:** Misma partitura de 3 notas.
- **Salida esperada:** Orden inverso de notas en recorrido hacia atrás.



```
✅ PASSED: ReversePlayback_ProducesNotesInReverseO
rder (4ms)
✅ PASSED: ReversePlayback_ProducesNotesInReverseO
rder (N/A)

========================================
Test Run Summary
========================================
Total Tests: 2
Passed: 2
Failed: 0
Skipped: 0
Result: ✅ SUCCEEDED
Test Execution Time: 0.26s (259ms)
Total Time (including build): 1.33s (1334ms)
========================================
```