

**TECNOLÓGICO DE MONTERREY**

TC2005B: Construcción de software y toma de decisiones



**Tecnológico  
de Monterrey**

## **Especificación de requerimientos de software**

Pablo Rocha Ojeda - A01028638

Luis Javier Karam - A01751941

Miguel Arriaga Velasco - A01028570

***Fecha de entrega*** - 01/04/2022

## Historias de usuarios

Historia de usuario #1 Instrumentos	
<b>Como</b> organización <b>quiero</b> un videojuego de instrumentos de percusión <b>para poder</b> educar sobre la percusión.	
<b>Validación:</b> <ul style="list-style-type: none"><li>● Mostrar demo al SF.</li><li>● Comprobar que el videojuego sea útil para sus miembros.</li></ul>	<b>Prioridad:</b> 10 <b>Estimación:</b> 300h

Historia de usuario #2 Página web	
<b>Como</b> usuario <b>quiero</b> una página web con el juego <b>para poder</b> tener fácil acceso al juego	
<b>Validación:</b> <ul style="list-style-type: none"><li>● Comprobar que el juego funcione de manera correcta dentro de la página</li><li>● Comprobar que todos los datos se muestren de manera correcta</li></ul>	<b>Prioridad:</b> 10 <b>Estimación:</b> 10h

Historia de usuario #3 Recolectar Información	
<b>Como</b> organización <b>quiero</b> recolectar información del progreso de los jugadores <b>para poder</b> ver su progreso	
<b>Validación:</b> <ul style="list-style-type: none"><li>● Información de las sesiones debe ser recolectada</li><li>● La información debe ser útil</li></ul>	<b>Prioridad:</b> 8 <b>Estimación:</b> 12h

Historia de usuario #4 Vista de administrador	
<b>Como</b> administrador <b>quiero</b> poder visualizar la información	

<b>para poder</b> interpretarla y ajustar la base de datos.	
<b>Validación:</b> <ul style="list-style-type: none"> <li>● Hacer login como administrador.</li> <li>● Visualizar la base de datos.</li> <li>● Verificar que los datos sean correctos.</li> </ul>	<b>Prioridad:</b> 7 <b>Estimación:</b> 5h

Historia de usuario #5 Juego informativo	
<b>Como organización</b> <b>quiero</b> educar a las personas sobre instrumentos percusivos <b>para poder</b> crear consciencia sobre instrumentos percusivos	
<b>Validación:</b> <ul style="list-style-type: none"> <li>● Comprobar que el contenido sea correcto</li> <li>● Comprobar que el contenido tenga valor real</li> </ul>	<b>Prioridad:</b> 8 <b>Estimación:</b> 1h

Historia de usuario #6 Clasificación	
<b>Como organización</b> <b>quiero</b> un videojuego apto para todas las edades <b>para poder</b> inspirar a jóvenes y adultos por igual a aprender sobre la percusión.	
<b>Validación:</b> <ul style="list-style-type: none"> <li>● Mostrar un demo al SF.</li> <li>● Que alumnos de todas las edades lo jueguen.</li> <li>● Pedirles su opinión.</li> </ul>	<b>Prioridad:</b> 2 <b>Estimación:</b> 1h

Historia de usuario #7 Tres Niveles	
<b>Como organización</b> <b>quiero</b> 3 niveles <b>para poder</b> tener un juego de dificultad progresiva.	
<b>Validación:</b> <ul style="list-style-type: none"> <li>● Existen 3 niveles</li> <li>● Los niveles son progresivos</li> </ul>	<b>Prioridad:</b> 8 <b>Estimación:</b> 18h

--	--

Historia de usuario #8 Lenguajes de Programación	
<b>Como</b> equipo <b>quiero</b> utilizar Unity, Javascript, NodeJS y MySQL <b>para poder</b> distribuir un juego de calidad.	
<b>Validación:</b> <ul style="list-style-type: none"> <li>• Se utiliza Unity</li> <li>• Se utiliza JS</li> <li>• Se utiliza SQL</li> </ul>	<b>Prioridad:</b> 4 <b>Estimación:</b> 80h

Historia de usuario #9 Login	
<b>Como</b> usuario <b>quiero</b> poder crear mis credenciales de inicio de sesión <b>para poder</b> almacenar mi información	
<b>Validación:</b> <ul style="list-style-type: none"> <li>• Crear un usuario</li> <li>• Iniciar sesión con sus credenciales</li> </ul>	<b>Prioridad:</b> 6 <b>Estimación:</b> 4h

Historia de usuario #10 Arte constante	
<b>Como</b> organización <b>quiero</b> que el juego tenga un estilo de arte unificado <b>para poder</b> mejorar la jugabilidad del juego	
<b>Validación:</b> <ul style="list-style-type: none"> <li>• Que todos los niveles y pantallas del juego tengan el mismo estilo visual</li> </ul>	<b>Prioridad:</b> 1 <b>Estimación:</b> 12h

Historia de usuario #11 Plataforma principal pc	
<b>Como</b> organización <b>quiero</b> que el juego se pueda ejecutar en pc <b>para poder</b> distribuir el juego a la mayor cantidad de personas posibles	

<b>Validación:</b> <ul style="list-style-type: none"> <li>Comprobar que el juego se ejecute de manera correcta en pc</li> </ul>	<b>Prioridad:</b> 10 <b>Estimación:</b> 2h
---	---

Historia de usuario #12 Single player	
<b>Como organización</b> <b>quiero</b> un juego de un solo jugador <b>para poder</b> medir el progreso de cada quien individualmente.	
<b>Validación:</b> <ul style="list-style-type: none"> <li>No se puede jugar de más jugadores</li> </ul>	<b>Prioridad:</b> 7 <b>Estimación:</b> 0h

Historia de usuario #13 Progreso registrado	
<b>Como jugador</b> <b>quiero</b> guardar el estado actual de mi juego <b>para poder</b> ver mi progreso.	
<b>Validación:</b> <ul style="list-style-type: none"> <li>Empezar una nueva sesión.</li> <li>Salir del juego.</li> <li>Volver a entrar al juego.</li> <li>Validar que se guarde el progreso.</li> </ul>	<b>Prioridad:</b> 5 <b>Estimación:</b> 20h

Historia de usuario # 14 Historia del juego	
<b>Como organización</b> <b>quiero</b> que se cuente la historia del juego <b>para poder</b> tener un juego congruente	
<b>Validación:</b> <ul style="list-style-type: none"> <li>La historia del juego se cuenta a través de sí mismo</li> </ul>	<b>Prioridad:</b> 1 <b>Estimación:</b> 5h

Historia de usuario # 15 Sin Hardware Externo	
---	--

<b>Como organización</b> <b>quiero</b> que no se necesite hardware externo <b>para poder</b> jugarlo solo con la computadora	
<b>Validación:</b> <ul style="list-style-type: none"> <li>El juego se puede jugar solo con la computadora</li> </ul>	<b>Prioridad:</b> 1 <b>Estimación:</b> 0h

Historia de usuario # 16 Base de datos	
<b>Como organización</b> <b>quiero</b> que la información se guarde en una base de datos <b>para poder</b> tener registro de los jugadores y sesiones.	
<b>Validación:</b> <ul style="list-style-type: none"> <li>La información se guarda en una base de datos de manera segura.</li> </ul>	<b>Prioridad:</b> 10 <b>Estimación:</b> 12h

## Product backlog

Requerimiento de usuario: Crear un juego musical en unity que colecciona información y le enseñe al jugador las bases de la percusión.

### Requerimientos funcionales:

1. Crear un sistema que convierta canciones a sus partituras.
2. Hacer que unity lea las partituras y genere las notas dinámicamente.
3. Crear inputs de diferentes teclas con las cuales el jugador pueda tocar la partitura.
4. Crear un sistema para que se generen diferentes notas dependiendo de su longitud.
5. Crear un sistema para mostrar la nota en el lugar adecuado de la partitura.
6. Sincronizar la música del juego con las notas en pantalla.
7. Generar un esquema de la base de datos.
8. Normalizar la base de datos a su tercera forma normal.
9. Crear un api con Node Js.
10. Conectar la base de datos al API.
11. Crear endpoints para poder agregar información del juego a la base de datos.
12. Crear endpoints para poder obtener información de la base de datos.
13. Crear una pantalla de inicio en donde el usuario haga login.
14. Crear un sistema que guarde el estado del juego actual.
15. Crear sistema para establecer el estado del juego actual, dependiendo de la información de un usuario registrado.

16. Crear una ventana administrativa donde se vea información relevante de los jugadores.
17. Crear una página web simple.
18. Insertar el juego a la página web.
19. Mandar los datos de Unity (high-scores, login, etc.) a la página web.

Requerimientos no funcionales:

20. Generar un fondo para el juego.
21. Generar sonidos para el juego.
22. Crear arte para el personaje.
23. Crear animaciones para el juego.
24. Crear arte para el ambiente (árboles, pantanos, monstruos, etc.)
25. Mostrar leaderboard dependiendo de los puntajes del juego, en la página web.
26. Crear una historia para el juego.

**Casos de uso UML**

Link:

[https://drive.google.com/file/d/1fw1fh4GsFW3W6OVMM372a9XNNlQvyU\\_8/view?usp=sharing](https://drive.google.com/file/d/1fw1fh4GsFW3W6OVMM372a9XNNlQvyU_8/view?usp=sharing)

Tarjetas para cada caso de uso:

<b>Generar notas de canciones</b>	
Descripción del caso de uso / Use case description detail	Con la ayuda de MIDI y UNITY se generarán las notas de las canciones en el videojuego.
Requerimientos relacionados / Related requirements	1,2,4,5,6
Objetivo en contexto / Goal in context	Generar las notas adecuadas para cada canción.
Precondiciones / Preconditions	Un archivo tipo MIDI debe de haberse generado y metido dentro del proyecto.
Condición final exitosa / Successful end condition	Las notas de la canción deben de estar coordinadas con la música. Asimismo, sus tiempos y tipos deben de corresponder a la partitura.
Condición final fallida / Failed end condition	Las notas no se generan o se generan incorrectamente y no acorde al archivo MIDI.
Actores primarios / Primary actors	<ul style="list-style-type: none"> <li>● MIDI</li> </ul>

	<ul style="list-style-type: none"> <li>• Unity</li> </ul>
Actores secundarios / Secondary actors	Audio de la música
Disparador / Trigger	El nivel comienza
Flujo principal / Main flow	<ol style="list-style-type: none"> <li>1. El jugador comienza un nivel.</li> <li>2. UNITY lee el archivo MIDI</li> <li>3. UNITY genera las notas correspondientes en pantalla.</li> </ol>
Extensiones / Extensions	<ol style="list-style-type: none"> <li>1. UNITY lee el archivo</li> <li>2. UNITY genera las notas de manera incorrecta.</li> </ol>

Generar inputs de las teclas	
Descripción del caso de uso / Use case description detail	Creación de diversos inputs para diferentes acciones dentro del videojuego.
Requerimientos relacionados / Related requirements	3
Objetivo en contexto / Goal in context	Que el jugador sea capaz de interactuar con el juego por medio de teclas. Si el jugador toca la tecla adecuada en el momento correcto se le agregan los puntos.
Precondiciones / Preconditions	<ul style="list-style-type: none"> <li>• El jugador debe de estar en un nivel.</li> <li>• Debe de haber notas en la pantalla.</li> </ul>
Condición final exitosa / Successful end condition	Si el jugador toca la nota adecuada en el momento adecuado se le deberían de dar puntos al jugador.
Condición final fallida / Failed end condition	Las teclas no crean ningún cambio en le juego.
Actores primarios / Primary actors	<ul style="list-style-type: none"> <li>• Unity</li> <li>• Jugador</li> </ul>
Actores secundarios / Secondary actors	Midi parser
Disparador / Trigger	El jugador presiona una tecla.
Flujo principal / Main flow	<ol style="list-style-type: none"> <li>1. Jugador está en un nivel</li> <li>2. La partitura sale en la pantalla.</li> <li>3. La nota llega al lugar adecuado.</li> <li>4. El jugador presiona la tecla.</li> </ol>



	5. La nota correspondiente desaparece. 6. Se dan los puntos adecuados.
Extensiones / Extensions	1. El jugador presiona la tecla. 2. Nada pasa.

Login del usuario	
Descripción del caso de uso / Use case description detail	Cuando se inicia el juego aparece una pantalla para hacer login y recuperar su estado anterior. Si no está registrado se registra.
Requerimientos relacionados / Related requirements	13
Objetivo en contexto / Goal in context	Que el jugador pueda ingresar al juego varias veces y se tenga un registro de su estado.
Precondiciones / Preconditions	El jugador debe estar registrado.
Condición final exitosa / Successful end condition	Al ingresar al juego el jugador es capaz de meter sus datos de juego e ingresar.
Condición final fallida / Failed end condition	<ul style="list-style-type: none"> <li>El jugador no puede ingresar al juego</li> <li>No se requiere login</li> </ul>
Actores primarios / Primary actors	<ul style="list-style-type: none"> <li>Usuario</li> <li>Base de datos</li> </ul>
Actores secundarios / Secondary actors	API
Disparador / Trigger	El jugador inicia el juego
Flujo principal / Main flow	1. El jugador inicia el juego. 2. Pantalla de login aparece. 3. Jugador ingresa su datos. 4. Si los datos son correctos ingresa a su perfil.
Extensiones / Extensions	1. El jugador inicia el juego. 2. El juego va directo a la pantalla de niveles o no abre.

<b>Guardar estado actual del juego</b>	
Descripción del caso de uso / Use case description detail	El juego debe de ser capaz de guardar el progreso del jugador para que continúe desde donde se quedó.
Requerimientos relacionados / Related requirements	14, 15
Objetivo en contexto / Goal in context	Que el jugador pueda guardar y observar su progreso, una mejor experiencia de usuario y educativa.
Precondiciones / Preconditions	<ul style="list-style-type: none"> <li>● Haber hecho login</li> <li>● Haber hecho progreso en el juego</li> </ul>
Condición final exitosa / Successful end condition	El jugador recupera su progreso y puede continuar desde donde se quedó.
Condición final fallida / Failed end condition	El progreso del jugador es eliminado y debe de comenzar desde el principio.
Actores primarios / Primary actors	<ul style="list-style-type: none"> <li>● Jugador</li> <li>● Unity</li> <li>● Base de datos de usuarios</li> </ul>
Actores secundarios / Secondary actors	<ul style="list-style-type: none"> <li>● API</li> </ul>
Disparador / Trigger	<ul style="list-style-type: none"> <li>● El jugador inicia el juego con un game file previo.</li> </ul>
Flujo principal / Main flow	<ol style="list-style-type: none"> <li>1. El jugador inicia el juego.</li> <li>2. El jugador selecciona un game file previo.</li> <li>3. El jugador puede jugar desde su progreso previo.</li> </ol>
Extensiones / Extensions	<ol style="list-style-type: none"> <li>1. El jugador selecciona un game file previo.</li> <li>2. El juego comienza desde cero.</li> </ol>

<b>Creación de assets</b>	
Descripción del caso de uso / Use case description detail	Se debe crear todo el arte visual y auditivo del juego para tener una experiencia placentera.
Requerimientos relacionados / Related requirements	20,21,22,23,24,25,26

Objetivo en contexto / Goal in context	Crear un estilo de arte uniforme alrededor de todos los niveles.
Precondiciones / Preconditions	Historia y temática del juego planteada
Condición final exitosa / Successful end condition	El arte es agradable y uniforme alrededor de todo el juego.
Condición final fallida / Failed end condition	Faltan personajes, fondos o otro tipo de arte/el arte tiene estilos diferentes.
Actores primarios / Primary actors	Desarrollador
Actores secundarios / Secondary actors	Usuario
Disparador / Trigger	El jugador inicia el juego
Flujo principal / Main flow	<ol style="list-style-type: none"> <li>1. Se hace una temática.</li> <li>2. Se crean los personajes.</li> <li>3. Se crean las notas.</li> <li>4. Se crea el fondo.</li> <li>5. Se une al juego.</li> </ol>
Extensiones / Extensions	<ol style="list-style-type: none"> <li>1. No se crean algunos personajes.</li> <li>2. El estilo de juego cambia entre niveles.</li> </ol>

<b>Crear ventana de administración</b>	
Descripción del caso de uso / Use case description detail	Se requiere de una ventana de administración para el product owner, organización o equipo de desarrolladores.
Requerimientos relacionados / Related requirements	16
Objetivo en contexto / Goal in context	Que se pueda observar las tendencias de la base de datos e interpretarla para obtener información.
Precondiciones / Preconditions	<ul style="list-style-type: none"> <li>• Que la base de datos sea existente y poblada.</li> <li>• Que el API pueda obtener información (GET) de la base de datos.</li> <li>• Que la página web esté construida.</li> </ul>
Condición final exitosa / Successful end condition	Que el administrador pueda observar diferentes aspectos e información de la base

	de datos.
Condición final fallida / Failed end condition	Que la base de datos no se pueda interpretar/observar.
Actores primarios / Primary actors	<ul style="list-style-type: none"> <li>● Base de datos de usuarios</li> <li>● Product owner</li> </ul>
Actores secundarios / Secondary actors	Desarrollador
Disparador / Trigger	Se accede a la página como administrador.
Flujo principal / Main flow	<ol style="list-style-type: none"> <li>1. Se accede a la página como administrador.</li> <li>2. Se obtiene información desde la base de datos.</li> <li>3. Se interpreta la información.</li> </ol>
Extensiones / Extensions	<ol style="list-style-type: none"> <li>1. Se accede a la página como administrador.</li> <li>2. No se puede visualizar información desde la base de datos o esta es incorrecta.</li> </ol>

<b>Conectar la base de datos al API</b>	
Descripción del caso de uso / Use case description detail	Conectar la base de base de datos al API para poder modificar las tablas con GET PUT, DELETE.
Requerimientos relacionados / Related requirements	10
Objetivo en contexto / Goal in context	Poder modificar la base de datos con llamadas al API para así guardar información del juego.
Precondiciones / Preconditions	<ul style="list-style-type: none"> <li>● Base de datos hecha y normalizada.</li> <li>● API creado.</li> </ul>
Condición final exitosa / Successful end condition	Se pueden hacer varias llamadas al API que afectan la base de datos y son capaces de : traer información, cambiar información, crear información y borrar información.
Condición final fallida / Failed end condition	El API no logra afectar la información de las bases de datos.
Actores primarios / Primary actors	<ul style="list-style-type: none"> <li>● Base de datos</li> </ul>

	<ul style="list-style-type: none"> <li>● Desarrollador</li> </ul>
Actores secundarios / Secondary actors	Página Web
Disparador / Trigger	El jugador hace una acción en el juego que requiera modificar la base de datos.
Flujo principal / Main flow	<ol style="list-style-type: none"> <li>1. El jugador hace una acción que requiera modificar la base de datos o requiera información de la base de datos. Ej: login</li> <li>2. Se hace la llamada API correspondiente.</li> <li>3. Los cambios se reflejan en el juego</li> <li>4. Los cambios se reflejan en la base de datos.</li> </ol>
Extensiones / Extensions	<ol style="list-style-type: none"> <li>1. El jugador no es capaz de extraer información de la base de datos.</li> <li>2. Sale algún error en la llamada del API.</li> </ol>

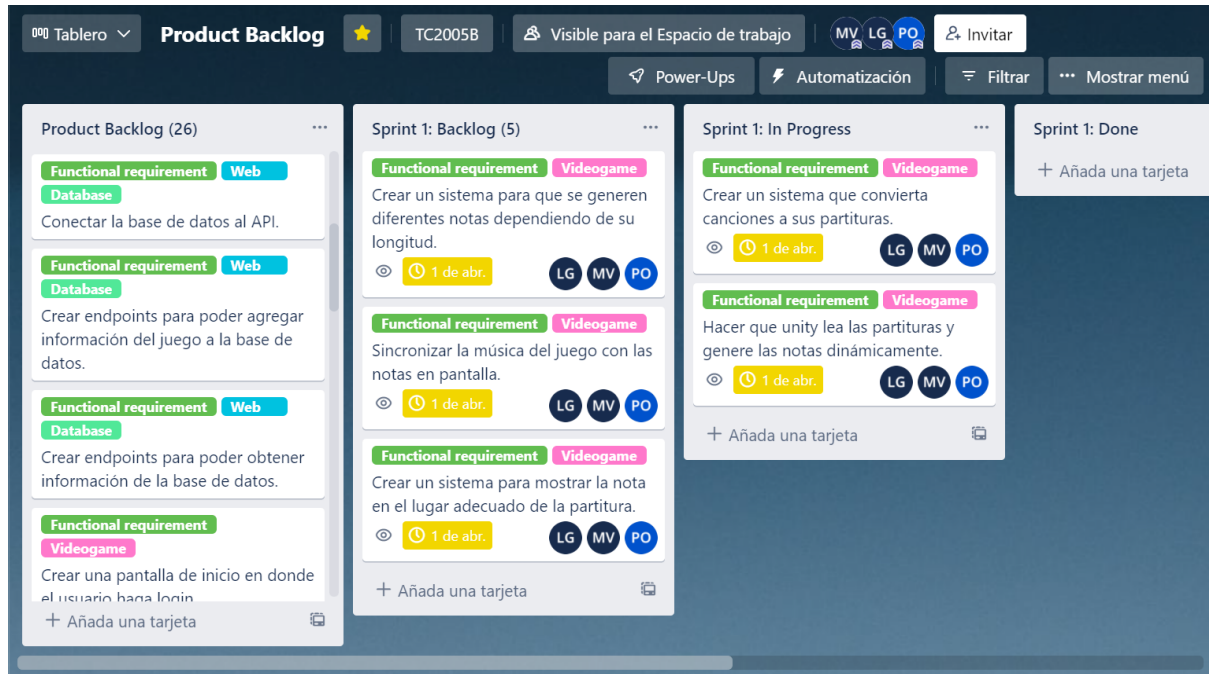
<b>Crear una página web simple con el juego</b>	
Descripción del caso de uso / Use case description detail	El juego debe de estar dentro de una página web.
Requerimientos relacionados / Related requirements	17, 18, 19
Objetivo en contexto / Goal in context	Que el juego pueda ser jugado por la mayor cantidad de jugadores posibles, que sea de fácil acceso.
Precondiciones / Preconditions	<ul style="list-style-type: none"> <li>● Crear el videojuego.</li> <li>● Exportar el videojuego.</li> <li>● Tener un dominio/host para la página.</li> </ul>
Condición final exitosa / Successful end condition	Que el videojuego se pueda jugar desde una página web.
Condición final fallida / Failed end condition	Que el videojuego no se pueda jugar desde la página web
Actores primarios / Primary actors	<ul style="list-style-type: none"> <li>● Desarrollador</li> <li>● Videojuego</li> </ul>
Actores secundarios / Secondary actors	API

	Base de datos
Disparador / Trigger	Exportar el videojuego como WebGL
Flujo principal / Main flow	<ol style="list-style-type: none"> <li>1. Se crea el videojuego.</li> <li>2. Se crea la página web.</li> <li>3. Se exporta el videojuego a la página web.</li> </ol>
Extensiones / Extensions	<ol style="list-style-type: none"> <li>1. No se exporta correctamente el videojuego a la página web.</li> <li>2. No se puede jugar el juego.</li> </ol>

<b>Crear un API con NodeJS</b>	
Descripción del caso de uso / Use case description detail	Se creará un backend del videojuego que servirá como intermedio entre nuestra base de datos y frontend.
Requerimientos relacionados / Related requirements	10,11,12
Objetivo en contexto / Goal in context	Que nuestro frontend sea capaz de comunicarse con la base de datos y realizar las acciones requeridas.
Precondiciones / Preconditions	Tener una base de datos completa y normalizada.
Condición final exitosa / Successful end condition	Se crea un servidor con Node Js con varias rutas capaz de modificar la base de datos.
Condición final fallida / Failed end condition	<ul style="list-style-type: none"> <li>● No se crea el servidor.</li> <li>● Alguna ruta no funciona adecuadamente.</li> </ul>
Actores primarios / Primary actors	Desarrollador
Actores secundarios / Secondary actors	Base de datos
Disparador / Trigger	Se acaba la formación de base de datos.
Flujo principal / Main flow	<ol style="list-style-type: none"> <li>1. Se crea un proyecto de Node JS sencillo.</li> <li>2. Se hacen rutas para modificar usuarios.</li> <li>3. Se hacen rutas para modificar clases.</li> <li>4. Se hacen rutas para modificar sesiones.</li> </ol>

## Tablero virtual

Link: <https://trello.com/b/UmyvtmKg>



## Sprint backlog #1

1. Crear un sistema que convierta canciones a sus partituras.
2. Hacer que unity lea las partituras y genere las notas dinámicamente.
4. Crear un sistema para que se generen diferentes notas dependiendo de su longitud.
5. Crear un sistema para mostrar la nota en el lugar adecuado de la partitura.
6. Sincronizar la música del juego con las notas en pantalla.