

## IHC — Proyecto Final

**Integrantes:**

Dávila Venturo, Fabio Eduardo

Niño Castañeda, Jesus Valentín

Casquino Paz, Daniel Ignacio

**Curso:** Interacción Humano Computador — CS4053

**Docente:** Chambilla Aquino, Teófilo

**Fecha de presentación:** 19 de Octubre, 2025

Universidad de Ingeniería y Tecnología (UTEC)

# 1 Motivación

El mixing es una actividad artística popular en el ámbito de la música electrónica, y consiste en combinar distintas canciones y/o sonidos para crear una pista única. Si bien esta actividad requiere ciertas habilidades técnicas y artísticas, también suele ser necesario contar con el equipo adecuado, el cual puede ser costoso y difícil de conseguir en ciertos contextos.

En la actualidad, aplicaciones como VirtualDJ proporcionan una solución digital para el mixing, y Tribe XR lleva esta idea un paso más allá, al ofrecer una experiencia en realidad virtual. Sin embargo, ambas aplicaciones funcionan bajo un sistema de suscripción. Por otro lado, si bien VirtualDJ ofrece una modalidad de pago único por \$299, esta puede ser una barrera de entrada alta para usuarios casuales.

Con el objetivo de satisfacer esta necesidad, proponemos el desarrollo de un simulador de DJ en realidad virtual. Este simulador permitirá a los usuarios aprender y desarrollar sus habilidades de mixing en un entorno inmersivo, sin la necesidad de contar con un equipo de mixing físico.

## **2 Requisitos de diseño y persona**

### **2.1 Requerimientos Funcionales**

Al funcionar como un simulador de DJ en realidad virtual, la aplicación debe contar con funcionalidades que simulan un equipo verdadero. Para lograrlo, proponemos los siguientes requerimientos funcionales.

N°	Descripción	Tipo
RF01	El usuario puede interactuar físicamente con el equipo de DJ (mezcladora, platos, perillas, etc.)	Must have
RF02	El usuario puede reproducir y pausar música con botones en la mesa de DJ	Must have
RF03	El usuario puede cambiar las canciones de los tracks a elección (precargadas o importables)	Must have
RF04	El usuario puede incrementar y disminuir el volumen de las canciones	Must have
RF05	El usuario puede cambiar y transicionar entre canciones usando un crossfader	Must have
RF06	El usuario puede aplicar filtros, como low pass, high pass, reverb, EQ, etc	Must have
RF07	El usuario puede mezclar múltiples canciones a la vez	Should have
RF08	El usuario puede ajustar el tono y la velocidad de las canciones	Should have
RF09	La aplicación debe tener un tutorial que explique las funcionalidades básicas	Must have
RF10	El usuario puede grabar sus mezclas	Nice to have
RF11	El setup de DJ debe incluir distintos parlantes, los cuales pueden ser conectados a distintas mesas de mezcla	Nice to have
RF12	El usuario puede girar los discos para controlar las canciones	Nice to have
RF13	La aplicación puede detectar el BPM de las canciones	Nice to have
RF14	El usuario puede utilizar samples y loops que se ajustan al BPM de las canciones	Nice to have
RF15	El usuario puede cambiar el BPM de las canciones en tiempo real	Nice to have
RF16	El usuario puede elegir distintos ambientes para desempeñar su set de DJ	Nice to have
RF17	El usuario puede afectar al público dependiendo de la calidad de su set (ej. animaciones, reacciones)	Nice to have

Table 1: Requerimientos funcionales

## 2.2 Requerimientos no Funcionales

De forma similar, planteamos requerimientos no funcionales, los cuales ayudarán a mejorar la experiencia de usuario.

N°	Descripción	Tipo
RNF01	La aplicación debe ejecutarse a un mínimo de 60 FPS	Must have
RNF02	La aplicación no debe tener retrasos perceptibles en el feedback de las acciones	Must have
RNF03	La aplicación debe ser utilizable sin experiencia previa como DJ	Should have

Table 2: Requerimientos no funcionales

## 2.3 Audiencia

La audiencia objetivo de la aplicación son adolescentes y adultos jóvenes de entre 14 y 30 años con un gusto por la música y el mixing. Estas personas suelen tener mayor afinidad por géneros musicales como, como EDM, house, etc., los cuales también son comúnmente utilizados en eventos sociales. Por último, los usuarios en este rango de edad suelen tener experiencia previa con tecnología y videojuegos, lo cual puede facilitar su adaptación a los controles de una aplicación VR.

## 2.4 Especificación de Usuario Final (User Person)

- **Nombre:** Alejandro Casquino
- **Edad:** 27 años
- **Ocupación:** Estudiante de arquitectura
- **Habilidad tecnológica:** Nivel intermedio/alto. Suele jugar videojuegos, con o sin experiencia previa en VR.
- **Intereses:** Fiestas, música electrónica, festivales, tecnología
- **Motivación:** Aprender a mezclar música y divertirse con una aplicación de realidad virtual.
- **Frustraciones:** Dificil acceso a equipos de DJ profesionales
- **Comportamiento:** Juega videojuegos regularmente, tanto móviles como en PC. Asiste a eventos musicales de forma regular, y tiene curiosidad por la tecnología VR.

## 2.5 Objetivo de la Experiencia

Al finalizar el desarrollo de la aplicación, nuestro objetivo es obtener un simulador de DJ en realidad virtual que permita a los usuarios sin experiencia previa en DJ aprender habilidades básica de mixing y divertirse con la música en un entorno immersivo.

## 2.6 Narrativa

Al ponerse el Meta Quest 2, Alejandro se encuentra en el centro de una cabina de DJ iluminada por luces de colores neón, al frente de un público virtual que baila al ritmo de la música. Frente a él, la mesa brilla con botones, vinilos y sliders. Aunque está solo en su habitación, la realidad virtual lo transporta a un festival, donde puede sentir el bajo de los parlantes y las voces del público.

La experiencia comienza cuando Alejandro elige su primera canción y la mezcla con otra, aplicando filtros y efectos. Cada cambio que hace modifica la música de forma inmediata. Gracias a los controles fáciles de usar, se puede explorar nuevas técnicas de mixing, girar los discos y ajustar el volumen como si estuviera en la cabina, sintiendo la emoción propia de un evento de verdad.

A medida que la sesión avanza, Alejandro prueba diferentes géneros musicales. Además, un tutorial le ayuda a dominar nuevas funciones, permitiéndole aprender nuevas técnicas. Luego de unas horas, Alejandro no sólo aprendió a mezclar música, sino que ha vivido una experiencia inmersiva y divertida desde su habitación.

## 2.7 Arco de la Historia

- **Introducción:** El usuario abre la aplicación, y se encuentra en el backstage de un festival musical. Se escucha la música de fondo y la multitud afuera.
- **Inicio:** El usuario se dirige al escenario principal, donde se encuentra la cabina de DJ. Al entrar, un breve tutorial le da la bienvenida y explica las funciones principales.
- **Adaptación:** El usuario selecciona su primera canción y experimenta moviendo los controles para mezclarla y aplicar efectos. La multitud virtual reacciona, animando al usuario a seguir.
- **Desarrollo:** El usuario comienza a experimentar con diferentes técnicas de mezcla, utilizando los efectos y controles disponibles en la cabina de DJ.
- **Clímax:** El usuario se siente completamente inmerso en la experiencia, como si estuviera en un verdadero festival de música. Las luces y el público brillan y saltan al ritmo de la música.
- **Fin:** El usuario termina su sesión de mezcla, luego de grabar su pista final. Finalmente, regresa al backstage, acompañado de música que disminuye gradualmente.

## 2.8 Métrica HEART

Métrica	Descripción	Indicador
Happiness	Encuestas dentro de la aplicación, así como medición de movimiento de los controles y la cámara	Calificación de satisfacción y movimiento frecuente de controles
Engagement	Cantidad de tiempo que los usuarios pasan en la aplicación, y la cantidad de canciones distintas utilizadas en el mix	Tiempo de uso medio y variedad de canciones
Adoption	Número de personas que utilizan la aplicación por primera vez	Nuevos usuarios por mes
Retention	Número de usuarios que prueban la aplicación en un periodo de una semana luego del uso inicial	Usuarios recurrentes por semana
Task Success	El usuario debe poder aplicar al menos 3 efectos diferentes sin problemas durante una sesión	Porcentaje de éxito mayor al 90%

Table 3: Métrica HEART

## 3 Framework de Diseño

El framework de diseño utilizado se basa en un proceso iterativo de planteamiento, prototipado, implementación, y evaluación. Primero, identificamos las necesidades y funcionalidades necesarias mediante una lluvia de ideas y análisis de mesas de DJ reales/virtuales. Luego, creamos un prototipo digital 2D para visualizar la interfaz y las posibles interacciones. Luego de tener una vista amplia de lo que se necesita implementar, el desarrollo se realiza en Unity, utilizando el SDK de interacción VR de Meta. Finalmente, una etapa de playtesting y crítica nos permite identificar áreas de mejora, las cuales se revisan en la siguiente etapa de desarrollo.

Con el objetivo proveer un UX óptimo y tener una ruta de diseño clara, aplicamos los principios de diseño que fueron discutidos en clases.

### 3.1 Principios de diseño

- **Signifiers:** Todos los elementos con los cuales el usuario debería poder interactuar están claramente indicados por colores, íconos, o formas que se utilizarán frecuentemente en el lenguaje visual de la aplicación. Con esto, pretendemos que el usuario pueda identificar rápidamente las funcionalidades disponibles, y acelerar el tiempo de aprendizaje.
- **Feedback:** Cada acción que el usuario realiza tendrá feedback auditivo (opcionalmente, visual). Girar perillas producirá “clicks”, los sliders desplazados por el usuario emitirán sonidos con un tono proporcional a su posición, y los botones sonarán y brillarán al ser presionados.
- **Mapping:** Los controles físicos de la mesa de mixing estarán configurados para seguir una dirección natural. Por ejemplo, una perilla en sentido horario incrementará el valor que controla (y viceversa). De forma similar, un slider que se mueva hacia arriba incrementará su valor.
- **Constraints:** Los controles como sliders y perillas tendrán límites físicos que impidan el movimiento. Al alcanzarlos, feedback visual y auditivo comunicará al usuario esta limitación.

### 3.2 Heurísticas de Usabilidad

- **Visibilidad del estado del sistema:** La aplicación siempre mostrará información sobre lo que está ocurriendo. Por ejemplo, al cambiar de canción, se reproducirá un sonido de “scratch” característico, y la imagen del vinilo cambiará. Una barra de progreso también mostrará la posición del playback de la canción.
- **Coherencia y estándares:** El lenguaje visual de la aplicación será consistente en todas los botones, perillas, y slider, y usarán íconos similares a los que se ven en mesas de DJ reales. Además, toda interfaz compartirá el mismo diseño visual y paleta de colores.
- **Prevención de errores:** Los vinilos que caigan fuera de la mesa de DJ serán teletransportados automáticamente a su posición original, evitando que el usuario pierda los vinilos.
- **Recuperación de errores:** Esta heurística será utilizada de forma escasa, dada la naturaleza física de los controles. Ejemplos particulares incluyen el brillo en rojo del botón de reproducción si no hay un vinilo colocado en la mesa.



- **Ayuda y documentación:** La aplicación tendrá un tutorial que enseñará al usuario las funcionalidades básicas. Este tutorial será opcional, y se podrá acceder a él desde el menú principal.

### 3.3 ¿Qué pueden lograr los usuarios?

Los usuarios podrán vivir la experiencia de ser un DJ en un entorno VR. Las funcionalidades incluyen:

- Reproducción de música en tiempo real.
- Interacción con controles como perillas y sliders.
- Aplicación de efectos y filtros a las canciones.
- Reproducción de samples y loops complementarios.
- Grabación y reproducción de mezclas.

### 3.4 Límites técnicos

- El uso de hand tracking impide el uso del feedback háptico de los controles, incrementando exponencialmente la importancia del feedback visual y auditivo.
- La precisión de la detección de BPM depende de los algoritmos implementados, y puede impactar la calidad de las mezclas, loops, y el rendimiento de la aplicación. Este último aspecto es importante, ya que la fluidez de la experiencia es indispensable en una aplicación VR.
- La implementación de parlantes y distintas mesas de DJ requiere de una arquitectura de software robusta y altamente modular, lo cual incrementa los tiempos de desarrollo.

### 3.5 Profundidad de presencia

Uno de los objetivos de la aplicación es maximizar la sensación de estar en un festival musical. Para lograrlo, se emplearán assets y materiales que creen un entorno colorido y dinámico, el cual se adapta a la música que el usuario está mezclando. El uso de efectos visuales y auditivos, como luces y el sonido del público, buscan maximizar la presencia del usuario en el entorno.

## 4 Prototipo

El objetivo principal del juego es que el usuario (el DJ) mezcle música en tiempo real para mantener a la audiencia virtual entretenida o para practicar en solitario.

## 4.1 Controles y Mapeo 1:1

El usuario usa controles de movimiento de VR. Sus manos virtuales, visibles en la imagen, imitan los movimientos de sus manos reales. La retroalimentación háptica (vibración) simula la sensación de tocar los discos (para el scratching) o sentir el “clic” de los botones y faders.

## 4.2 Mecánicas de Mezcla

El usuario interactúa directamente con el equipo en la cabina:

- **Platos (Turntables):** Agarrar físicamente el disco virtual para hacer scratch.
- **Sliders:** Ajustar el pitch (tono) y el tempo (velocidad).
- **Controles básicos:** Dar “play/pausa” y “cue”.
- **Mezcladora (Mixer):** Usar los faders de volumen para las canciones, el crossfader para cambiar entre platos, y las perillas de ecualización (EQ) para ajustar bajos, medios y agudos.
- **Efectos (FX):** Activar y manipular perillas para añadir efectos como reverb, delay o filtros, tal como se haría en un equipo real.

## 4.3 El Bucle del Juego (Game Loop)

### 4.3.1 Biblioteca de Música

El usuario tendría una lista de canciones (virtual) para elegir o subir sus propios archivos.

### 4.3.2 Práctica en Solitario

El usuario tendrá la opción de practicar en un estudio en solitario, un ambiente sin presión para practicar todo lo que quiera.

### 4.3.3 Reacción del Público

El usuario también podrá tocar frente a un público en una discoteca. La audiencia (visible al frente) reaccionaría dinámicamente. Si el DJ hace una buena transición, mantiene el ritmo y usa efectos de forma atractiva, el público bailará más, gritará y la energía del lugar subirá.

### 4.3.4 Retroalimentación

Si el DJ comete errores (mala mezcla, silencio, transiciones fallidas), el público dejará de bailar, empezará a abuchear o incluso a irse.

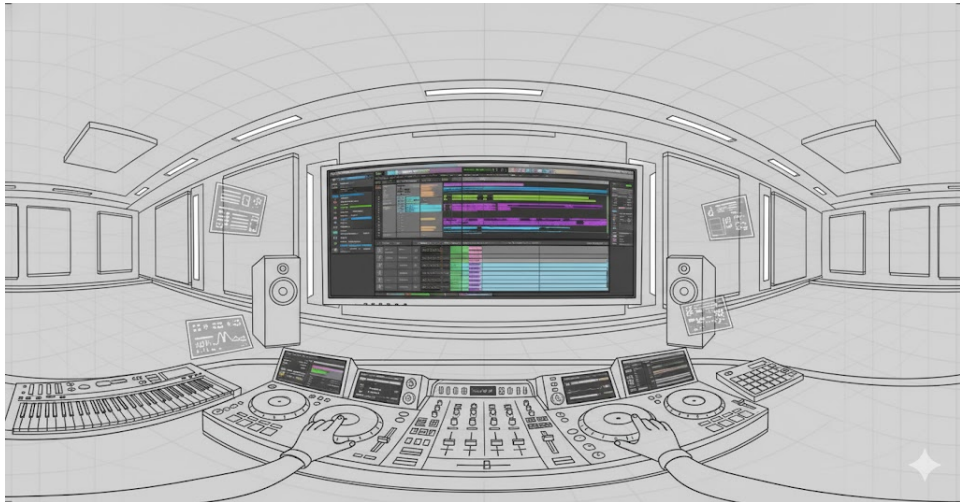
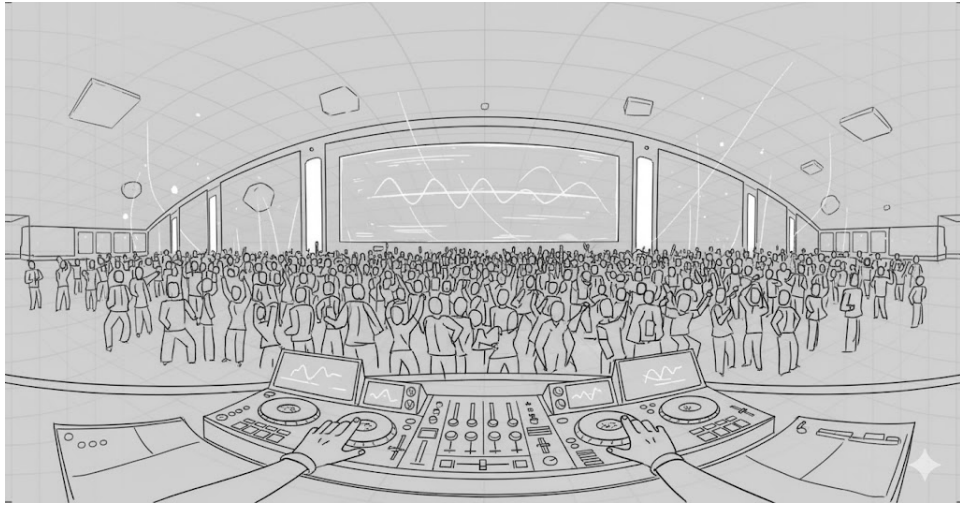


Figure 1: Interfaz y visualización del escenario (dj-demo1 y dj-demo2).

## 5 Cómo navegaría el usuario (Navegación y UI)

### 5.1 Navegación en el Menú Principal (Fuera de la actuación)

Al iniciar el juego, el usuario estaría en el estudio.

#### 5.1.1 Puntero Láser

El usuario usaría sus controles como punteros láser para apuntar y seleccionar opciones en un menú 2D flotante.

### **5.1.2 Opciones**

Aquí seleccionaría modos de juego (“Discoteca” para pasar al escenario, “Sesión de práctica” para quedarse en el estudio), podría personalizar su equipo (elegir diferentes decks o mezcladoras) o accedería a tutoriales.

## **5.2 Navegación en el Juego**

Mientras se está en el escenario, no se puede pausar para buscar en un menú. Durante la práctica en el estudio, se puede hacer de todo sin restricciones.

### **5.2.1 Pantallas de los Decks**

Las pantallas pequeñas integradas en los decks (visibles en la imagen) serían la interfaz principal. El usuario usaría perillas o touchpads en el deck virtual para navegar por su biblioteca de canciones.

### **5.2.2 Gestos de “Agarrar”**

Para cargar una canción, el usuario la seleccionaría en la pantalla y luego usaría un gesto de “agarrar” (apretando el gatillo del control) para “arrastrarla” al plato virtual deseado.

### **5.2.3 Interfaz Holográfica/Muñeca**

Para salir de la discoteca, el usuario podría mirar su smartwatch virtual. Ahí, podría presionar un botón para salir y terminar el juego prematuramente.

## **6 Cronograma propuesto**

### **6.1 Entregable 1 (15% - Semana 11)**

#### **6.1.1 Requerimientos Desarrollados**

- RF01: interacción básica con placeholders
- RF02: botón de play presionable (sin efecto)
- RF04: fader de volumen (muestra valor, pero no controla sonido todavía)

#### **6.1.2 Objetivo**

Tener un entorno de prueba en el Quest 2 donde el usuario puede ver sus manos e interactuar con un objeto del equipo de DJ.

### 6.1.3 Proyecto y VR

- Crear proyecto en Unity (URP - Universal Render Pipeline).
- Configurar el Meta XR SDK.
- Implementar el rig de VR: El usuario puede ver sus manos (controladores).

### 6.1.4 Modelos y Entorno Básico

- Importar modelos 3D placeholder para la mezcladora y los platos.
- Crear la “Escena de Prueba” para alinear la altura y escala del equipo de DJ.

### 6.1.5 Interacción Núcleo

- Implementar la interacción física básica: El usuario puede “tocar” los placeholders del equipo y sus controles, pero sin tener funcionalidad aún.
- Botones de *play* presionables presentes en el modelo, mas no reproducen música todavía.
- Script para un solo componente: Un fader de volumen que se puede agarrar y mover, y que imprime su valor en la consola (ej. 0.8).

### 6.1.6 Resultado esperado

El usuario está en una sala vacía, ve sus manos y puede mover un fader de volumen.

## **6.2 Entregable 2 (30% - Semana 12)**

### **6.2.1 Requerimientos De**

- RF02: Botón de *play* funcional para reproducir/pausar música.
- RF03: Carga básica de canciones (opciones disponibles predefinidas).
- RF04: Fader de volumen que controla el volumen real de la canción.

### **6.2.2 Objetivo**

Implementar el sistema de audio. El usuario debe poder mezclar dos canciones usando el equipo básico.

### **6.2.3 Sistema de Carga de Audio**

- Implementar la lógica de carga dinámica de Ogg Vorbis.
- Crear una UI temporal (menú 2D simple) para seleccionar 2 canciones de una lista de 10.
- Asignar las canciones a los AudioSources de “Plato A” y “Plato B”.

### **6.2.4 Mezcla Básica**

- Conectar el fader de volumen del Entregable 1 al AudioSource.volume.
- Habilitar el botón de *play* para que reproduzca/pausa la canción en el AudioSource correspondiente.
- Implementar el Crossfader: Un script que controla el volumen de ambos platos simultáneamente (sube A mientras baja B).
- Capacidad limitada de selección de canciones (lista de canciones limitada, sin carga de archivos externos).

### **6.2.5 Resultado esperado**

El usuario puede cargar dos canciones y mezclarlas usando los faders de volumen y el crossfader. La mecánica de audio principal está probada.

## **6.3 Entregable 3 (50% - Semana 13)**

### **6.3.1 Requerimientos Desarrollados**

- RF01: Interacción avanzada con el deck (scratching en platos, perillas de EQ, botones de rewind(cue)).
- RF06: Aplicar efectos básicos (EQ: Bajos, Medios, Agudos).
- RF16: Primer ambiente implementado (Estudio).

### **6.3.2 Objetivo**

Completar el primer ambiente y la funcionalidad completa del deck. El prototipo debe ser jugable en modo solitario.

### **6.3.3 Ambiente 1: Estudio**

- Importar los assets 3D finales para el estudio en solitario.
- Implementar la iluminación.

### **6.3.4 Interacción Avanzada**

- Implementar el resto de interacciones del deck:
  - **Platos (Turntables):** Lógica de Scratching (vincular la velocidad al AudioSource.pitch y time).
  - **Perillas (Knobs):** Scripts para las perillas de ecualización (EQ: Bajos, Medios, Agudos).
  - **Botones:** Botones de “Cue” para rewind de canciones.

### **6.3.5 Resultado esperado**

Un prototipo funcional. El usuario está en un estudio detallado y tiene control total sobre el equipo de DJ (mezcla, EQs, scratching).

## **6.4 Entregable 4 (70% - Semana 14)**

### **6.4.1 Requerimientos**

- RF16: Segundo ambiente implementado (Discoteca) con audiencia dinámica básica.
- RF17: Sistema de audiencia básico (reacciona a si la música está sonando o no).

### **6.4.2 Objetivo**

Implementar el segundo ambiente y la audiencia.

### **6.4.3 Ambiente 2: Discoteca**

- Modelar e importar los assets 3D para la Discoteca.

### **6.4.4 Sistema de Audiencia**

- Crear o importar modelos 3D low-poly para el público (siluetas o modelos simples tipo Mii).
- Implementar un script de audiencia básico: El público tiene dos animaciones (Idle y Bailando).
- Lógica simple: Si el AudioSource principal está sonando, el público “Baila”. Si no, Idle.

### **6.4.5 Resultado esperado**

El usuario puede elegir entre el Estudio y la Discoteca. La discoteca tiene un público que reacciona a la música.



## **6.5 Entregable 5 (100% - Semana 16)**

### **6.5.1 Requerimientos**

- RF01: Interacción completa y pulida con el equipo de DJ.
- RF06: Se añade un efecto extra (filtro low/high pass).
- RF16: Menú principal sólido para seleccionar ambiente.
- RF17: Sistema de audiencia mejorado (reacciona a la calidad de la mezcla).

### **6.5.2 Objetivo**

Presentar un prototipo estable, pulido y completo.

### **6.5.3 Pulido Avanzado (15%)**

- Mejorar el sistema de audiencia (ej. que reaccionen a la mezcla o al crossfader).
- Añadir un feature de audio extra (ej. un filtro “low/high pass” en la mezcladora).
- Crear un menú principal sólido para seleccionar el ambiente.

### **6.5.4 Bug Fixing (Arreglo de errores)**

- Corregir errores con las manos.
- Corregir problemas de carga de audio.
- Ajustar la iluminación, reducir polígonos y optimizar scripts que consuman mucha CPU.

### **6.5.5 Usabilidad**

- Ajustar la “sensibilidad” de las perillas y el scratching.
- Implementar retroalimentación háptica (vibración) en los controles al tocar botones o hacer scratch.

### **6.5.6 Resultado esperado**

Un prototipo completo, optimizado y pulido.