SwipeMath AR

Implementación de Gráficos 3D Web con Three.js y Realidad Aumentada

Computación Gráfica - Práctica Calificada 5

Sergio Pezo J. **Arbues Perez V.** **André Pacheco T.**
Jared Orihuela C.

July 4, 2025

FACULTAD DE CIENCIAS

Table of Contents

Objetivos de la Práctica Calificada 5

- **Desarrollar un juego de 1 escenario (hipercasual)**

Objetivos de la Práctica Calificada 5

- **Desarrollar un juego de 1 escenario (hipercasual)**
- **Asociado ODS-Educación** (Objetivo de Desarrollo Sostenible 4)

Objetivos de la Práctica Calificada 5

- **Desarrollar un juego de 1 escenario (hipercasual)**
- **Asociado ODS-Educación** (Objetivo de Desarrollo Sostenible 4)
- **Debe hacer uso de Realidad Aumentada**

Objetivos de la Práctica Calificada 5

- **Desarrollar un juego de 1 escenario (hipercasual)**
- **Asociado ODS-Educación** (Objetivo de Desarrollo Sostenible 4)
- **Debe hacer uso de Realidad Aumentada**

Nuestra Solución: SwipeMath AR

Un juego educativo de matemáticas en Realidad Aumentada diseñado para niños de 6-8 años

Problema: Educación Matemática Temprana

- Los métodos tradicionales pueden ser poco motivadores

Problema: Educación Matemática Temprana

- Los métodos tradicionales pueden ser poco motivadores
- Falta de interactividad y retroalimentación inmediata

Problema: Educación Matemática Temprana

- Los métodos tradicionales pueden ser poco motivadores
- Falta de interactividad y retroalimentación inmediata
- Dificultad para mantener la atención de niños pequeños

Problema: Educación Matemática Temprana

- Los métodos tradicionales pueden ser poco motivadores
- Falta de interactividad y retroalimentación inmediata
- Dificultad para mantener la atención de niños pequeños
- Brecha entre el aprendizaje abstracto y la experiencia tangible

Problema: Educación Matemática Temprana

- Los métodos tradicionales pueden ser poco motivadores
- Falta de interactividad y retroalimentación inmediata
- Dificultad para mantener la atención de niños pequeños
- Brecha entre el aprendizaje abstracto y la experiencia tangible

Oportunidad

La Realidad Aumentada permite crear experiencias educativas inmersivas que combinan el mundo físico con elementos digitales interactivos

Table of Contents

SwipeMath AR: Concepto del Juego

2 Descripción del Juego

- Juego hipercasual de matemáticas

SwipeMath AR: Concepto del Juego

2 Descripción del Juego

- Juego hipercasual de matemáticas
- Usa marcadores AR físicos para la interacción

SwipeMath AR: Concepto del Juego

2 Descripción del Juego

- Juego hipercasual de matemáticas
- Usa marcadores AR físicos para la interacción
- El jugador mueve una tarjeta para controlar a Stitch

SwipeMath AR: Concepto del Juego

2 Descripción del Juego

- Juego hipercasual de matemáticas
- Usa marcadores AR físicos para la interacción
- El jugador mueve una tarjeta para controlar a Stitch
- Resuelve operaciones matemáticas rápidamente

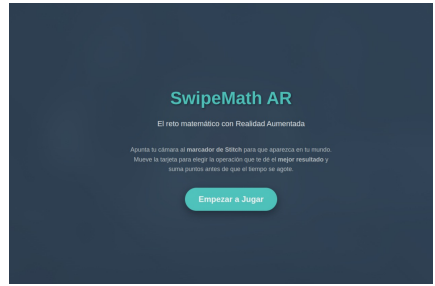
SwipeMath AR: Concepto del Juego

- Juego hipercasual de matemáticas
- Usa marcadores AR físicos para la interacción
- El jugador mueve una tarjeta para controlar a Stitch
- Resuelve operaciones matemáticas rápidamente
- Sistema de puntuación y niveles progresivos

SwipeMath AR: Concepto del Juego

2 Descripción del Juego

- Juego hipercasual de matemáticas
- Usa marcadores AR físicos para la interacción
- El jugador mueve una tarjeta para controlar a Stitch
- Resuelve operaciones matemáticas rápidamente
- Sistema de puntuación y niveles progresivos



Mecánica de Juego

2 Descripción del Juego

- **Inicio**: El jugador apunta la cámara al marcador AR

Mecánica de Juego

2 Descripción del Juego

- **Inicio**: El jugador apunta la cámara al marcador AR
- **Visualización**: Aparece Stitch en 3D sobre la tarjeta

Mecánica de Juego

2 Descripción del Juego

- **Inicio**: El jugador apunta la cámara al marcador AR
- **Visualización**: Aparece Stitch en 3D sobre la tarjeta
- **Desafío**: Se muestran dos operaciones matemáticas

Mecánica de Juego

2 Descripción del Juego

- **Inicio**: El jugador apunta la cámara al marcador AR
- **Visualización**: Aparece Stitch en 3D sobre la tarjeta
- **Desafío**: Se muestran dos operaciones matemáticas
- **Selección**: Mover la tarjeta izquierda/derecha para elegir

Mecánica de Juego

2 Descripción del Juego

- **Inicio**: El jugador apunta la cámara al marcador AR
- **Visualización**: Aparece Stitch en 3D sobre la tarjeta
- **Desafío**: Se muestran dos operaciones matemáticas
- **Selección**: Mover la tarjeta izquierda/derecha para elegir
- **Objetivo**:

Mecánica de Juego

2 Descripción del Juego

- **Inicio**: El jugador apunta la cámara al marcador AR
- **Visualización**: Aparece Stitch en 3D sobre la tarjeta
- **Desafío**: Se muestran dos operaciones matemáticas
- **Selección**: Mover la tarjeta izquierda/derecha para elegir
- **Objetivo**:
  - Suma/Multiplicación: Elegir el resultado mayor

Mecánica de Juego

2 Descripción del Juego

- **Inicio**: El jugador apunta la cámara al marcador AR
- **Visualización**: Aparece Stitch en 3D sobre la tarjeta
- **Desafío**: Se muestran dos operaciones matemáticas
- **Selección**: Mover la tarjeta izquierda/derecha para elegir
- **Objetivo**:
  - Suma/Multiplicación: Elegir el resultado mayor
  - Resta/División: Elegir la menor penalización

Mecánica de Juego

2 Descripción del Juego

- **Inicio**: El jugador apunta la cámara al marcador AR
- **Visualización**: Aparece Stitch en 3D sobre la tarjeta
- **Desafío**: Se muestran dos operaciones matemáticas
- **Selección**: Mover la tarjeta izquierda/derecha para elegir
- **Objetivo**:
  - Suma/Multiplicación: Elegir el resultado mayor
  - Resta/División: Elegir la menor penalización

Retroalimentación Inmediata

Sonidos y animaciones indican respuestas correctas/incorrectas

Table of Contents

Stack Tecnológico

3 Arquitectura Técnica

Flask (Python)

Servidor Web

Stack Tecnológico

3 Arquitectura Técnica

Flask (Python)

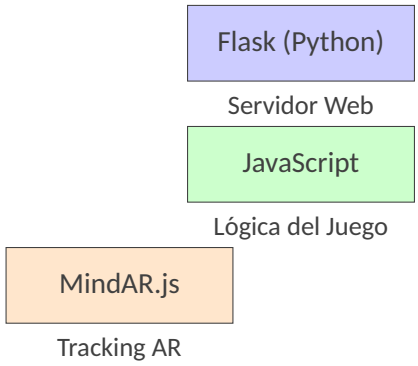
Servidor Web

JavaScript

Lógica del Juego

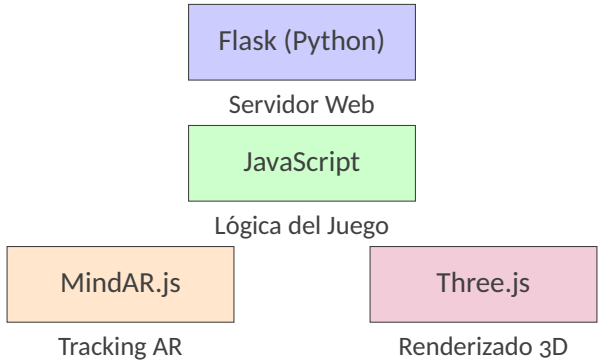
Stack Tecnológico

3 Arquitectura Técnica



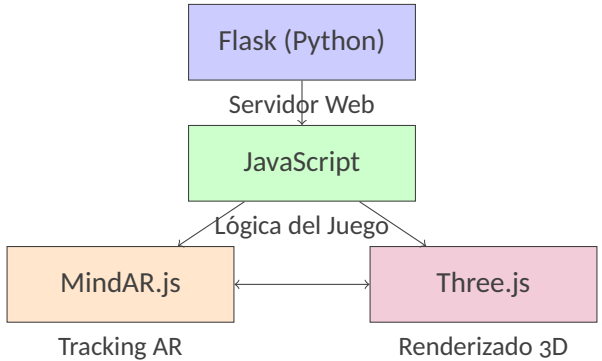
Stack Tecnológico

3 Arquitectura Técnica



Stack Tecnológico

3 Arquitectura Técnica



Arquitectura del Sistema con Three.js

3 Arquitectura Técnica

- **Capa de Renderizado 3D**:

- **Capa de Renderizado 3D**:
  - Three.js Scene Manager

Arquitectura del Sistema con Three.js

3 Arquitectura Técnica

- **Capa de Renderizado 3D**:
  - Three.js Scene Manager
  - WebGL Renderer

Arquitectura del Sistema con Three.js

3 Arquitectura Técnica

- **Capa de Renderizado 3D**:
  - Three.js Scene Manager
  - WebGL Renderer
  - Camera Controls

Arquitectura del Sistema con Three.js

3 Arquitectura Técnica

- **Capa de Renderizado 3D**:
  - Three.js Scene Manager
  - WebGL Renderer
  - Camera Controls
- **Integración AR**:

Arquitectura del Sistema con Three.js

3 Arquitectura Técnica

- **Capa de Renderizado 3D**:
  - Three.js Scene Manager
  - WebGL Renderer
  - Camera Controls
- **Integración AR**:
  - MindAR para tracking

Arquitectura del Sistema con Three.js

3 Arquitectura Técnica

- **Capa de Renderizado 3D**:
  - Three.js Scene Manager
  - WebGL Renderer
  - Camera Controls
- **Integración AR**:
  - MindAR para tracking
  - Three.js para visualización

Arquitectura del Sistema con Three.js

3 Arquitectura Técnica

- **Capa de Renderizado 3D**:
  - Three.js Scene Manager
  - WebGL Renderer
  - Camera Controls
- **Integración AR**:
  - MindAR para tracking
  - Three.js para visualización
  - Sincronización de matrices

Arquitectura del Sistema con Three.js

3 Arquitectura Técnica

- **Capa de Renderizado 3D**:
  - Three.js Scene Manager
  - WebGL Renderer
  - Camera Controls
- **Integración AR**:
  - MindAR para tracking
  - Three.js para visualización
  - Sincronización de matrices

Pipeline de Renderizado

1. Captura de video stream
2. Detección de marcador AR
3. Cálculo de pose 3D
4. Three.js transforma matrices
5. Renderizado WebGL
6. Composición final

Table of Contents

Three.js en SwipeMath AR

4 Implementación con Three.js

- **Motor de renderizado 3D**: Three.js maneja toda la visualización

Three.js en SwipeMath AR

4 Implementación con Three.js

- **Motor de renderizado 3D**: Three.js maneja toda la visualización
- **Integración con MindAR**: Sincronización de cámara y tracking

Three.js en SwipeMath AR

4 Implementación con Three.js

- **Motor de renderizado 3D**: Three.js maneja toda la visualización
- **Integración con MindAR**: Sincronización de cámara y tracking
- **Pipeline de renderizado**: Scene, Camera, Renderer

Three.js en SwipeMath AR

4 Implementación con Three.js

- **Motor de renderizado 3D**: Three.js maneja toda la visualización
- **Integración con MindAR**: Sincronización de cámara y tracking
- **Pipeline de renderizado**: Scene, Camera, Renderer
- **Gestión de recursos**: Modelos GLTF, texturas, materiales

Three.js en SwipeMath AR

4 Implementación con Three.js

- **Motor de renderizado 3D**: Three.js maneja toda la visualización
- **Integración con MindAR**: Sincronización de cámara y tracking
- **Pipeline de renderizado**: Scene, Camera, Renderer
- **Gestión de recursos**: Modelos GLTF, texturas, materiales

¿Por qué Three.js?

- WebGL simplificado
- Compatibilidad móvil/desktop
- Ecosistema robusto
- Rendimiento optimizado

Configuración de la Escena Three.js

4 Implementación con Three.js

```
// Inicializacion de Three.js con MindAR
const mindarThree = new MindARThree({
    container: arContainer,
    imageTargetSrc: '/static/assets/targets/targets.mind'
});

const {renderer, scene, camera} = mindarThree;

// Configuracion del renderer
renderer.setPixelRatio(window.devicePixelRatio);
renderer.setSize(window.innerWidth, window.innerHeight);
renderer.outputEncoding = THREE.sRGBEncoding;

// Iluminacion de la escena
const light = new THREE.HemisphereLight(0xffffff, 0xbbbbff, 1);
scene.add(light);

// Luz direccional para sombras
const directionalLight = new THREE.DirectionalLight(0xffffff, 0.8);
directionalLight.position.set(0, 10, 5);
scene.add(directionalLight);
```

Carga y Manipulación de Modelos 3D

4 Implementación con Three.js

```
// Carga del modelo GLTF con Three.js
import {GLTFLoader} from 'three/examples/jsm/loaders/GLTFLoader.js';

async function loadStitchModel() {
    const loader = new GLTFLoader();
    const gltf = await loader.loadAsync(
        '/static/assets/models/stitch/scene.gltf'
    );

    // Configurar el modelo
    stitch = gltf.scene;
    stitch.scale.set(0.2, 0.2, 0.2);
    stitch.rotation.y = Math.PI;

    // Animar el modelo con Three.js
    mixer = new THREE.AnimationMixer(stitch);
    if (gltf.animations.length > 0) {
        const action = mixer.clipAction(gltf.animations[0]);
        action.play();
    }

    return stitch;
}
```

Sistema de Detección de Posición 3D

4 Implementación con Three.js

```
// Raycasting para interaccion 3D
const raycaster = new THREE.Raycaster();
const mouse = new THREE.Vector2();

// Conversion de coordenadas AR a espacio 3D
function updateStitchPosition() {
    // Obtener posicion del marcador en el mundo
    const markerPosition = anchor.group.position;

    // Aplicar transformaciones Three.js
    stitch.position.copy(markerPosition);

    // Detectar lado para seleccion
    const worldPos = new THREE.Vector3();
    stitch.getWorldPosition(worldPos);

    // Proyectar a coordenadas de pantalla
    const screenPos = worldPos.project(camera);

    if (screenPos.x < -0.15) {
        highlightChoice('left');
    } else if (screenPos.x > 0.15) {
        highlightChoice('right');
    }
}
```

Renderizado y Animación con Three.js

4 Implementación con Three.js

```
// Loop de renderizado principal
function animate() {
    requestAnimationFrame(animate);

    // Actualizar animaciones del modelo
    if (mixer) {
        const delta = clock.getDelta();
        mixer.update(delta);
    }

    // Rotacion suave del personaje
    if (stitch && gameActive) {
        stitch.rotation.y += 0.01;

        // Efecto de flotacion
        stitch.position.y = Math.sin(Date.now() * 0.001) * 0.05;
    }

    // Actualizar elementos UI en 3D
    updateOperationLabels();

    // Renderizar escena
    renderer.render(scene, camera);
}

// Iniciar loop de animacion
animate();
```

Características Three.js Implementadas

4 Implementación con Three.js

- **Geometrías y Materiales**:

Características Three.js Implementadas

- **Geometrías y Materiales**:
  - PlaneGeometry para UI

Características Three.js Implementadas

- **Geometrías y Materiales**:
  - PlaneGeometry para UI
  - MeshBasicMaterial

Características Three.js Implementadas

- **Geometrías y Materiales**:
  - PlaneGeometry para UI
  - MeshBasicMaterial
  - TextureLoader

Características Three.js Implementadas

- **Geometrías y Materiales**:
  - PlaneGeometry para UI
  - MeshBasicMaterial
  - TextureLoader
- **Iluminación**:

Características Three.js Implementadas

- **Geometrías y Materiales**:
  - PlaneGeometry para UI
  - MeshBasicMaterial
  - TextureLoader
- **Iluminación**:
  - HemisphereLight

Características Three.js Implementadas

- **Geometrías y Materiales**:
  - PlaneGeometry para UI
  - MeshBasicMaterial
  - TextureLoader
- **Iluminación**:
  - HemisphereLight
  - DirectionalLight

Características Three.js Implementadas

- **Geometrías y Materiales**:
  - PlaneGeometry para UI
  - MeshBasicMaterial
  - TextureLoader
- **Iluminación**:
  - HemisphereLight
  - DirectionalLight
  - AmbientLight

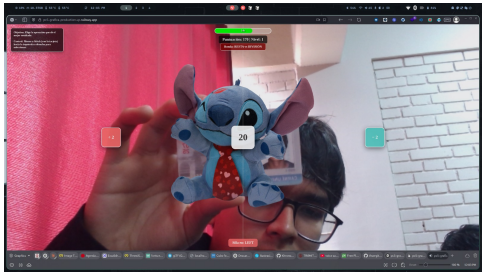
Características Three.js Implementadas

- **Geometrías y Materiales**:
  - PlaneGeometry para UI
  - MeshBasicMaterial
  - TextureLoader
- **Iluminación**:
  - HemisphereLight
  - DirectionalLight
  - AmbientLight
- **Animación**:
  - AnimationMixer
  - Clock para timing
  - RequestAnimationFrame
- **Optimización**:
  - Frustum culling
  - LOD (Level of Detail)
  - Texture compression

Table of Contents

Estados del Juego

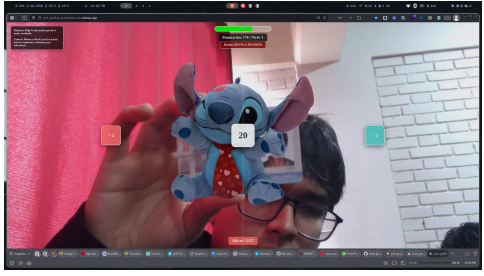
5 Resultados y Demostración



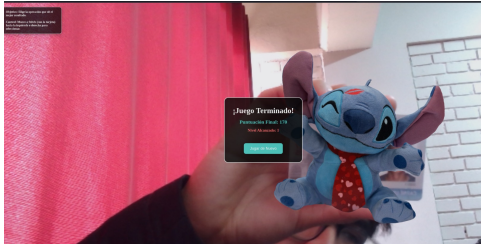
Juego en Acción

Estados del Juego

5 Resultados y Demostración



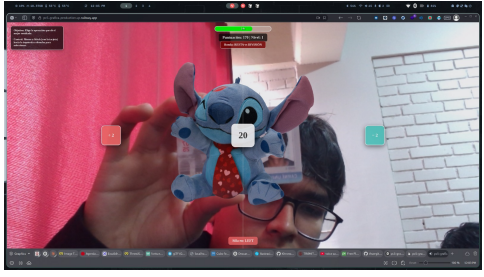
Juego en Acción



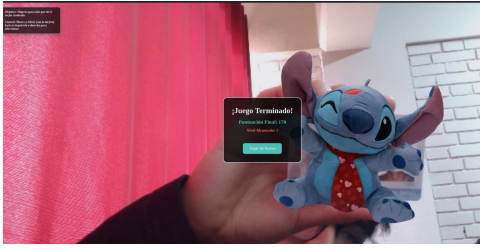
Fin del Juego

Estados del Juego

5 Resultados y Demostración



Juego en Acción



Fin del Juego

Sistema de Retroalimentación

- Efectos de sonido diferenciados
- Animaciones visuales
- Actualización inmediata de puntuación

Características Implementadas

- **Escenario Único** (requisito PC5)

Características Implementadas

5 Resultados y Demostración

- **Escenario Único** (requisito PC5)
- **Realidad Aumentada** funcional

Características Implementadas

5 Resultados y Demostración

- **Escenario Único** (requisito PC5)
- **Realidad Aumentada** funcional
- **ODS 4 - Educación de Calidad**

Características Implementadas

5 Resultados y Demostración

- **Escenario Único** (requisito PC5)
- **Realidad Aumentada** funcional
- **ODS 4 - Educación de Calidad**
- Sistema de dificultad progresiva

Características Implementadas

5 Resultados y Demostración

- **Escenario Único** (requisito PC5)
- **Realidad Aumentada** funcional
- **ODS 4 - Educación de Calidad**
- Sistema de dificultad progresiva
- Retroalimentación auditiva y visual

Características Implementadas

5 Resultados y Demostración

- **Escenario Único** (requisito PC5)
- **Realidad Aumentada** funcional
- **ODS 4 - Educación de Calidad**
- Sistema de dificultad progresiva
- Retroalimentación auditiva y visual
- Interfaz intuitiva para niños

Características Implementadas

- **Escenario Único** (requisito PC5)
- **Realidad Aumentada** funcional
- **ODS 4 - Educación de Calidad**
- Sistema de dificultad progresiva
- Retroalimentación auditiva y visual
- Interfaz intuitiva para niños

Métricas del Juego

- Tiempo promedio: 2-5 min
- Edad objetivo: 6-8 años
- Operaciones: +, -, ×, ÷

Table of Contents

6 Conclusiones

Logros Técnicos con Three.js

6 Conclusiones

- ✓ Renderizado 3D fluido en navegador web

Logros Técnicos con Three.js

- ✓ Renderizado 3D fluido en navegador web
- ✓ Integración exitosa Three.js + MindAR

Logros Técnicos con Three.js

6 Conclusiones

- ✓ Renderizado 3D fluido en navegador web
- ✓ Integración exitosa Three.js + MindAR
- ✓ Pipeline de gráficos optimizado para móviles

Logros Técnicos con Three.js

6 Conclusiones

- ✓ Renderizado 3D fluido en navegador web
- ✓ Integración exitosa Three.js + MindAR
- ✓ Pipeline de gráficos optimizado para móviles
- ✓ Gestión eficiente de recursos 3D (GLTF)

Logros Técnicos con Three.js

6 Conclusiones

- ✓ Renderizado 3D fluido en navegador web
- ✓ Integración exitosa Three.js + MindAR
- ✓ Pipeline de gráficos optimizado para móviles
- ✓ Gestión eficiente de recursos 3D (GLTF)
- ✓ Animaciones y efectos visuales en tiempo real

Logros Técnicos con Three.js

- ✓ Renderizado 3D fluido en navegador web
- ✓ Integración exitosa Three.js + MindAR
- ✓ Pipeline de gráficos optimizado para móviles
- ✓ Gestión eficiente de recursos 3D (GLTF)
- ✓ Animaciones y efectos visuales en tiempo real

Contribución a Computación Gráfica

Demostración práctica de cómo Three.js permite crear experiencias AR educativas complejas con rendimiento óptimo en la web

Trabajo Futuro

6 Conclusiones

- **Expansión de Contenido**:

Trabajo Futuro

- **Expansión de Contenido**:
  - Más tipos de operaciones matemáticas

Trabajo Futuro

6 Conclusiones

- **Expansión de Contenido**:
  - Más tipos de operaciones matemáticas
  - Problemas de lógica y geometría

Trabajo Futuro

6 Conclusiones

- **Expansión de Contenido**:
  - Más tipos de operaciones matemáticas
  - Problemas de lógica y geometría
- **Mejoras Técnicas**:

Trabajo Futuro

- **Expansión de Contenido**:
  - Más tipos de operaciones matemáticas
  - Problemas de lógica y geometría
- **Mejoras Técnicas**:
  - Soporte multijugador

Trabajo Futuro

- **Expansión de Contenido**:
  - Más tipos de operaciones matemáticas
  - Problemas de lógica y geometría
- **Mejoras Técnicas**:
  - Soporte multijugador
  - Analytics de aprendizaje

Trabajo Futuro

- **Expansión de Contenido**:
  - Más tipos de operaciones matemáticas
  - Problemas de lógica y geometría
- **Mejoras Técnicas**:
  - Soporte multijugador
  - Analytics de aprendizaje
  - Más modelos 3D y animaciones

Trabajo Futuro

- **Expansión de Contenido**:
  - Más tipos de operaciones matemáticas
  - Problemas de lógica y geometría
- **Mejoras Técnicas**:
  - Soporte multijugador
  - Analytics de aprendizaje
  - Más modelos 3D y animaciones
- **Características Sociales**:

Trabajo Futuro

6 Conclusiones

- **Expansión de Contenido**:
  - Más tipos de operaciones matemáticas
  - Problemas de lógica y geometría
- **Mejoras Técnicas**:
  - Soporte multijugador
  - Analytics de aprendizaje
  - Más modelos 3D y animaciones
- **Características Sociales**:
  - Tabla de puntuaciones

Trabajo Futuro

6 Conclusiones

- **Expansión de Contenido**:
  - Más tipos de operaciones matemáticas
  - Problemas de lógica y geometría
- **Mejoras Técnicas**:
  - Soporte multijugador
  - Analytics de aprendizaje
  - Más modelos 3D y animaciones
- **Características Sociales**:
  - Tabla de puntuaciones
  - Modo competitivo

SwipeMath AR

Implementación de gráficos 3D con Three.js
para educación matemática en AR

Demostrando el poder de Three.js para
crear experiencias educativas inmersivas en la web

Práctica Calificada 5 - Computación Gráfica

Repositorio: `https://github.com/thsergitox/pc5-grafica`