**Programa de Ingenieria de Software**

**Programación Orientada a Entornos Multimedia**

# Actividad Teórico–Práctica

**Objetivo**

Explorar las diferentes opciones que ofrece **Three.js** a través de **React con @react-three/fiber y drei**, describiendo los cambios realizados y representando procesos en el aplicativo dispuesto para prácticas.

El desarrollo de la práctica debe incluir:

* Descripciones personales del trabajo en binas (parejas).
* Publicación de los cambios en el repositorio personal de GitHub destinado para esta práctica.
* Clone el siguiente proyecto para iniciar con la actividad:

git clone https://github.com/guswill24/Portafolio\_guswill.git

**Productos a entregar**

* Documento con las respuestas y evidencias solicitadas.
* URL del repositorio de github, donde se desarrolló la práctica.

**Integrantes**

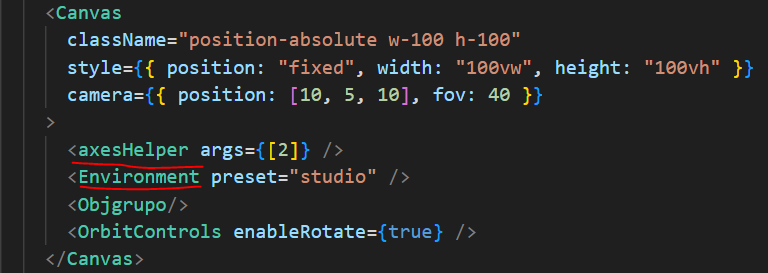
(Nombres de los estudiantes aquí)

# Actividades

### 1. Exploración de componentes en Ejercicio4

En el repositorio de práctica, dirígete al menú **Prácticas** y observa la animación **Agrupación de Objetos – React** ubicada en pages/Ejercicio4.jsx.

* Analiza el script, donde se está utilizando React con **@react-three/fiber** y **drei**.
* Describe qué hace cada componente marcado en rojo.



**Respuesta:**

* Explica qué otras opciones permite la propiedad **preset** del componente <Environment>.

**Respuesta:**

* Aplica al menos **3 valores diferentes de preset** en el proyecto y muestra evidencia de los resultados en este documento.

**Respuesta:**

### 2. Análisis del componente <Objgrupo />

En el mismo script, revisa el componente personalizado <Objgrupo /> y desarrolla las siguientes actividades:

1. **Explicación teórica**
   * Describe cómo se realiza la **agrupación de objetos** en React utilizando **@react-three/fiber** y **drei**.

**Respuesta:**

1. **Exploración de propiedades**
   * Investiga qué otras **props** soporta el componente <group>.

**Respuesta:**

1. **Práctica en el laboratorio**
   * En el menú **Laboratorio**, bajo el nombre **Practica-1**, realiza **tres cambios significativos** aplicando diferentes propiedades en <group>.
   * Cada agrupación debe incluir **3 objetos diferentes**.
2. **Rotaciones y observaciones**
   * Aplica la propiedad **rotation** en cada <group> utilizando los siguientes valores:
     + rotation={[8, 0, 0]}
     + rotation={[10, 0, 0]}
     + rotation={[15, 0, 0]}
   * Describe los efectos observados en cada caso.
3. **Entrega de resultados**
   * Refleja todos los cambios en el proyecto, dentro del menú **Laboratorio**, con el nombre: **Practica-1**.

### 3. Análisis de jerarquía de escena

Observa el siguiente esquema de una escena en React Three Fiber:

Scene

├── AmbientLight

├── DirectionalLight

├── Mesh (suelo)

└── Group

├── Mesh (cubo)

└── Mesh (esfera)

Responde:

1. ¿Qué elementos dependen directamente de la **escena principal** y cuáles dependen del **grupo**?  
   **Respuesta:**
2. ¿Qué ocurre si movemos el <group> en el eje Y?  
   **Respuesta:**
3. ¿El orden en que aparecen las luces y el piso en el JSX afecta la jerarquía de la escena? Justifica tu respuesta.  
   **Respuesta:**

### 4. Análisis de agrupación con Three.js (sin React Fiber)

En el menú **Prácticas**, dirígete al archivo pages/Ejercicio5.jsx y analiza el componente **ThreeJSGrupo**.

1. **Identificación de la agrupación**
   * Observe cómo se declara el objeto const group = new THREE.Group();.
   * Explique en sus palabras cuál es la función de THREE.Group() y cómo se diferencian los objetos agregados a la **escena** (scene.add(...)) frente a los objetos agregados al **grupo** (group.add(...)).
2. **Transformaciones del grupo**
   * Cree el script en el menú Laboratorio con el nombre: Practica-2, para aplicar a group las siguientes transformaciones:
     + group.rotation.y = Math.PI / 4;
     + group.scale.set(1.5, 1.5, 1.5);
     + group.position.set(0, 2, 0);
   * Documente cómo estas transformaciones afectan a los tres cubos al mismo tiempo.
3. **Comparación individual vs. agrupado**
   * Elimine temporalmente el group y agregue los cubos directamente a la escena (scene.add(cube1, cube2, cube3);).
   * Explique cuál es la diferencia al aplicar transformaciones a cada cubo por separado frente a aplicarlas al grupo.

### 5. Análisis de Luces en Three.js y React Three Fiber

**Objetivo:**

Explorar los diferentes tipos de luces disponibles en Three.js, comprender cómo afectan a los modelos y objetos en escena, y trasladar este análisis a React Three Fiber para evaluar similitudes, diferencias y optimización en el manejo de luces.

**Instrucciones:**

1. **Observación y análisis inicial**

Revise el script dado y realice un cuadro comparativo de los tipos de luces implementadas:

* AmbientLight
* DirectionalLight
* HemisphereLight
* PointLight
* SpotLight
* RectAreaLight

Para cada una describa:

* Color usado.
* Intensidad configurada.
* Propiedades adicionales (posición, ángulo, distancia, etc.).
* Efecto visual en los objetos de la escena.

1. **Experimentación en Three.js**

* Modifique los valores de intensidad, color y posición de al menos 3 tipos de luces.
* Tome capturas que muestren cómo cambia la percepción de las sombras y el realismo.

1. **Migración a React Three Fiber**

Realizar actividad en el menú **Laboratorio**, bajo el nombre **Practica-3**

* Replique esta misma escena en React Three Fiber.
* Para cada luz usada en Three.js, investigue y use su equivalente en R3F (<ambientLight />, <directionalLight />, <hemisphereLight />, <pointLight />, <spotLight />, <rectAreaLight />).
* Responda: ¿qué sintaxis cambia y qué ventajas ofrece R3F frente a la manipulación de luces en Three.js puro?

1. **Uso de lil-gui en Three.js**

Realizar actividad en el menú **Laboratorio**, bajo el nombre **Practica-4**

Instrucciones:

1. Siguiendo con el componente TiposLuces.jsx
2. Identifique la sección donde se crea el panel **lil-gui**:
3. const gui = new GUI();
4. gui.add(ambientLight, 'intensity').min(0).max(3).step(0.001);
   * Aquí ambientLight es la luz que ya estaba agregada a la escena.
   * "intensity" es la propiedad que se está controlando desde la interfaz.
   * .min(0).max(3).step(0.001) define el rango y precisión del slider.
5. Modifica el script para que no solo controles la luz ambiental, sino también:

* **El color de la luz ambiental.**
* **La intensidad de la luz puntual.**

De la siguiente forma:

|  |
| --- |
| // GUI - Control de parámetros  const gui = new GUI();  // Luz ambiental: intensidad y color  gui.add(ambientLight, "intensity", 0, 3, 0.1).name("Ambient Intensity");  gui.addColor({ color: ambientLight.color.getHex() }, "color")  .name("Ambient Color")  .onChange((value) => ambientLight.color.set(value));  // Luz puntual: intensidad  gui.add(pointLight, "intensity", 0, 5, 0.1).name("Point Intensity"); |

* Describe en tu palabras qué hace cada línea.

**Respuesta:**

* ¿Cómo se está vinculando la GUI con las propiedades de las luces?

**Respuesta:**

**Experimenta en la escena:**

* Cambia la **intensidad de la luz ambiental** entre 0 y 3.
* Modifica el **color de la luz ambiental** a rojo, verde, azul y blanco.
* Ajusta la **intensidad de la luz puntual** entre 0 y 5.

Presenta las evidencias frente a esos cambios

**Respuesta:**

### Análisis de Sombras y Luces

**Objetivo**: Comprender cómo se configuran las las sombras en Three.js, analizando las propiedades del script y usando la interfaz Lil-GUI para realizar cambios interactivos.

**Actividades**

1. **Análisis del Código [src\components\SombrasEscena.jsx]**
   * Describa qué significan las propiedades castShadow y receiveShadow en los objetos.
2. **Exploración con GUI**
   * Ajusta la intensidad de la AmbientLight y la DirectionalLight.
   * Cambia las posiciones del DirectionalLight con los controles del GUI.
   * Documenta en capturas cómo varían las sombras en cada caso.
3. **Ejercicio práctico**
   * Apaga todas las luces menos una (descomenta/comenta en el código).
   * Describe cómo se ve la escena con solo cada tipo de luz.
   * Propón cuál combinación da mayor realismo y justifica tu respuesta.

### ****Experimentación con Materiales y Sombras****

Realizar actividad en el menú **Laboratorio**, bajo el nombre **Practica-5**

**Objetivo**: Explorar cómo los materiales interactúan con las sombras y las luces.

### Actividades

1. Cambie el material de la esfera (MeshStandardMaterial) por MeshPhongMaterial y MeshLambertMaterial.
   * Explique la diferencia en cómo reflejan la luz y proyectan sombras.
2. Usa la GUI para modificar las propiedades de metalness y roughness.
   * Documente cómo afecta al aspecto de la esfera.
3. Agrega un segundo objeto (ejemplo: cubo) con el mismo material y compara cómo se ven bajo diferentes luces.

### Práctica: Exploración de la Escena Casa 3D

Realizar actividad en el menú **Laboratorio**, bajo el nombre **Practica-6**

**Objetivo**

Comprender cómo se construye una escena en Three.js desde React, identificando la relación entre **geometrías, materiales, texturas, luces, sombras y efectos ambientales** (cielo y niebla).

Usar la **GUI (lil-gui)** para experimentar con parámetros y reflexionar sobre cómo afectan la visualización.

### Instrucciones

1. **Analice la estructura del script CasaEscena.jsx:**
   * ¿Qué objetos principales conforman la escena (casa, arbustos, tumbas, fantasmas, cielo, suelo)?, identifique los nombres de los objetos.
   * ¿Qué papel cumplen los **grupos (THREE.Group)** en la organización de los elementos?
2. **Experimente con la GUI:**
   * Cambie los valores de floorDisplacementScale y floorDisplacementBias.
   * Observe cómo la textura del piso cambia en detalle y profundidad.
   * Responda: ¿qué efecto visual logra cuando aumentas mucho la escala de desplazamiento?
3. **Luces y Sombras:**
   * Identifique los tres tipos de luces usados: AmbientLight, DirectionalLight, PointLight.
   * Explique la diferencia en su comportamiento dentro de la escena.
   * Active/desactive el castShadow y receiveShadow en **walls** y **roof** para ver cómo cambian las sombras proyectadas.
4. **Explore el movimiento de los fantasmas:**
   * Observe cómo cambian sus posiciones a lo largo del tiempo.
   * ¿Qué función cumplen las fórmulas con Math.sin y Math.cos?
   * Proponga una variación en el movimiento (ej. hacer que un fantasma se mueva más rápido o en otra dirección).
5. **Ambiente:**
   * Modifique los parámetros de sky.material.uniforms (por ejemplo rayleigh, turbidity).
   * Active/desactive la FogExp2.
   * Reflexione: ¿cómo influyen el cielo y la niebla en la atmósfera de la escena?
6. **Propuesta propia:**
   * Reemplace las tumbas por cruces con material y textura propia. Para ello, cree dos cajas alargadas, agréguelas en un mismo grupo para formar la cruz y utilice este grupo para poblar el escenario, siguiendo el mismo procedimiento aplicado con las tumbas