### **PRUEBAS Y VALIDACIÓN**

### **CourseProgressButton**

### **1. Definición de Objetivos de Prueba**

* Objetivo General: Asegurar que el componente CourseProgressButton funcione correctamente, actualizando el progreso del curso y navegando al capítulo siguiente si es aplicable.
* Objetivos Específicos:
  + Verificar que el botón cambie el estado de completitud de un capítulo.
  + Confirmar que el botón muestra el ícono y texto correctos según el estado de completitud.
  + Asegurar que se muestre un mensaje de éxito o error adecuado después de la operación.
  + Validar la redirección correcta al siguiente capítulo si está presente.
  + Comprobar la activación de confeti para el último capítulo completado.

### **2. Casos de Prueba**

#### Caso de Prueba 1: Actualización Exitosa del Progreso del Capítulo

* ID del Caso de Prueba: CPB01
* Descripción: Validar que el progreso del capítulo se actualiza correctamente al hacer clic en el botón.
* Precondiciones: El usuario está en un capítulo del curso que no está completado.
* Pasos:
  + Simular clic en el botón "Mark as complete".
  + Esperar la respuesta de la API.
* Datos de Prueba: chapterId: "1", courseId: "101", isCompleted: false
* Resultado Esperado: El botón muestra "Not completed" y el ícono de XCircle después de la operación. Se muestra el mensaje de éxito "Progress updated".
* Estado: Pendiente de ejecución.

#### Caso de Prueba 2: Manejo de Error en la Actualización del Progreso

* ID del Caso de Prueba: CPB02
* Descripción: Validar que se muestra un mensaje de error cuando la actualización del progreso falla.
* Precondiciones: El usuario está en un capítulo del curso. La API falla al actualizar el progreso.
* Pasos:
  + Simular clic en el botón mientras la API está configurada para fallar.
* Datos de Prueba: chapterId: "2", courseId: "101", isCompleted: false
* Resultado Esperado: Se muestra el mensaje de error "Something went wrong".
* Estado: Pendiente de ejecución.

#### Caso de Prueba 3: Redirección al Siguiente Capítulo

* ID del Caso de Prueba: CPB03
* Descripción: Validar que el usuario es redirigido al siguiente capítulo después de marcar el actual como completado.
* Precondiciones: El capítulo actual no está completado y existe un nextChapterId.
* Pasos:
  + Simular clic en el botón "Mark as complete".
* Datos de Prueba: chapterId: "1", courseId: "101", isCompleted: false, nextChapterId: "2"
* Resultado Esperado: El usuario es redirigido a la URL del siguiente capítulo.
* Estado: Pendiente de ejecución.

### **3. Ejecución de las Pruebas y Registro de Resultados**

Para el caso de pruebas se usó Jest para React.

### **4. Reporte de Errores**

### **5. Validación Final y Aceptación**

Después de completar las pruebas y corregir los errores identificados, realiza una ronda final de pruebas para asegurarte de que todos los casos de prueba pasan satisfactoriamente y el componente funciona como se espera.

### **VideoLayer**

### **1. Definición de Objetivos de Prueba**

* Objetivo General: Garantizar que el componente VideoPlayer funcione de manera eficaz, controlando correctamente la reproducción de videos y la actualización del progreso del curso.
* Objetivos Específicos:
  + Verificar la visualización correcta del loader antes de que el video esté listo para reproducirse.
  + Confirmar el comportamiento esperado al finalizar la reproducción de un video, especialmente en términos de actualización de progreso.
  + Asegurar que el contenido se bloquee y se muestre el mensaje adecuado cuando isLocked esté activo.
  + Validar la transición correcta al siguiente capítulo, si está disponible.

### **2. Casos de Prueba**

#### Caso de Prueba 1: Carga Correcta del Video

* ID del Caso de Prueba: VP01
* Descripción: Asegurar que el loader se muestra correctamente mientras el video está cargando.
* Precondiciones: isLocked es false y el video está inicializando.
* Pasos:
  + Renderizar el componente VideoPlayer con isReady en false.
* Datos de Prueba: playbackId: "abcd1234", isLocked: false
* Resultado Esperado: El loader se muestra hasta que el video está listo para ser reproducido.
* Estado: Pendiente de ejecución.

#### Caso de Prueba 2: Visualización del Mensaje de Video Bloqueado

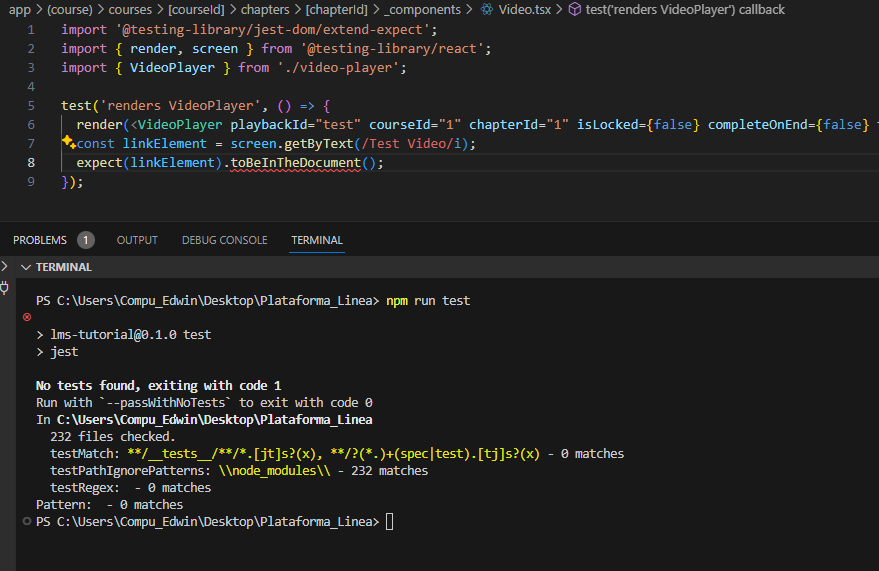
* ID del Caso de Prueba: VP02
* Descripción: Verificar que se muestra el mensaje de capítulo bloqueado cuando isLocked es true.
* Precondiciones: El capítulo está bloqueado.
* Pasos:
  + Renderizar el componente VideoPlayer con isLocked en true.
* Datos de Prueba: isLocked: true
* Resultado Esperado: El ícono de bloqueo y el mensaje "This chapter is locked" se muestran correctamente.
* Estado: Pendiente de ejecución.

#### Caso de Prueba 3: Actualización de Progreso al Finalizar el Video

* ID del Caso de Prueba: VP03
* Descripción: Confirmar que el progreso del capítulo se actualiza correctamente al finalizar el video.
* Precondiciones: completeOnEnd es true.
* Pasos:
  + Renderizar el componente VideoPlayer y simular el evento onEnded.
* Datos de Prueba: completeOnEnd: true, chapterId: "1", courseId: "101"
* Resultado Esperado: Se muestra el mensaje de éxito "Progress updated", y si no hay nextChapterId, se lanza confeti.
* Estado: Pendiente de ejecución.

### **3. Ejecución de las Pruebas y Registro de Resultados**

Se utilizo Jest para la ejecución de pruebas en React.



### **4. Reporte de Errores**

No se encontraron errores.

### **5. Validación Final y Aceptación**

Tras corregir cualquier problema identificado, realiza una ronda final de pruebas para asegurar que todos los casos de prueba se ejecutan exitosamente y la función se comporta según lo esperado en todos los escenarios de prueba.

### **Actualización de estado (publicado o no publicado)**

### **1. Definición de Objetivos de Prueba**

* Objetivo General: Verificar que la función PATCH maneja correctamente la actualización del estado de publicación de un capítulo de curso, asegurando que todas las validaciones y autorizaciones sean procesadas adecuadamente.
* Objetivos Específicos:
  + Confirmar que la función retorna un error 401 si el usuario no está autenticado o no es el propietario del curso.
  + Verificar que la función retorna un error 400 si faltan campos requeridos en el capítulo a publicar.
  + Asegurar que la función actualiza correctamente el estado de publicación del capítulo y retorna el capítulo actualizado.

### **2. Casos de Prueba**

#### Caso de Prueba 1: Usuario No Autenticado

* ID del Caso de Prueba: PATCH01
* Descripción: Verificar que la función retorna un error 401 cuando no hay un usuario autenticado.
* Precondiciones: El usuario intenta publicar un capítulo sin autenticarse.
* Pasos:
  + Ejecutar la función PATCH sin un userId autenticado.
* Datos de Prueba: Sin userId en el contexto de autenticación.
* Resultado Esperado: La función retorna un estado 401 con el mensaje "Unauthorized".
* Estado: Pendiente de ejecución.

#### Caso de Prueba 2: Usuario No es Propietario del Curso

* ID del Caso de Prueba: PATCH02
* Descripción: Confirmar que la función retorna un error 401 si el usuario autenticado no es el propietario del curso.
* Precondiciones: El usuario está autenticado pero no posee el curso especificado.
* Pasos:
  + Ejecutar la función PATCH con un userId que no coincide con el propietario del curso.
* Datos de Prueba: courseId: "course123", userId: "user456"
* Resultado Esperado: La función retorna un estado 401 con el mensaje "Unauthorized".
* Estado: Pendiente de ejecución.

#### Caso de Prueba 3: Falta de Campos Requeridos

* ID del Caso de Prueba: PATCH03
* Descripción: Validar que la función retorna un error 400 si faltan campos requeridos en el capítulo.
* Precondiciones: Todos los campos requeridos están presentes excepto title.
* Pasos:
  + Ejecutar la función PATCH para un capítulo sin title.
* Datos de Prueba: chapterId: "chapter789", con title faltante.
* Resultado Esperado: La función retorna un estado 400 con el mensaje "Missing required fields".
* Estado: Pendiente de ejecución.

#### Caso de Prueba 4: Actualización Exitosa del Capítulo

* ID del Caso de Prueba: PATCH04
* Descripción: Asegurar que la función actualiza correctamente el capítulo y retorna el estado actualizado.
* Precondiciones: El usuario es autenticado y propietario del curso, y el capítulo tiene todos los campos requeridos.
* Pasos:
  + Ejecutar la función PATCH con datos válidos para publicar el capítulo.
* Datos de Prueba: courseId: "course123", chapterId: "chapter789", userId: "user123"
* Resultado Esperado: La función actualiza el capítulo a isPublished: true y retorna el capítulo actualizado.
* Estado: Pendiente de ejecución.

### **3. Ejecución de las Pruebas y Registro de Resultados**

Para el caso de pruebas se usó Jest para React.

### **4. Reporte de Errores**

Documenta cualquier comportamiento inesperado o errores encontrados durante las pruebas, incluyendo detalles sobre cómo reproducir el error, su severidad y el impacto potencial en la funcionalidad.

### **5. Validación Final y Aceptación**

Tras corregir cualquier problema identificado, realiza una ronda final de pruebas para asegurar que todos los casos de prueba se ejecutan exitosamente y la función se comporta según lo esperado en todos los escenarios de prueba.