**Historia de Usuario**

**Como usuario del sistema (profesor, estudiante u otro rol con acceso),  
quiero poder visualizar las respuestas que los estudiantes han dado a las distintas actividades de un Learning Path,  
para entender su progreso, la calidad de sus respuestas, y así tener una visión clara del desempeño de los alumnos independientemente de mi rol.**

**Esta historia de usuario surge del requerimiento de extender la visibilidad de las respuestas de los estudiantes, que anteriormente estaba restringida a los profesores. Con esta modificación, cualquier usuario con acceso a la pestaña de "Respuestas Estudiantes" puede revisar las respuestas, facilitando el monitoreo y análisis del aprendizaje.**

**Roles Involucrados**

1. **Profesor:**
   * **Antes: Podía crear Learning Paths, agregar actividades y visualizar el progreso de los estudiantes.**
   * **Después: Continúa con las mismas capacidades, pero ahora la vista de respuestas está disponible para cualquier usuario sin depender exclusivamente de su rol de profesor.**
2. **Estudiante:**
   * **Antes: Podía ver solo su propio progreso (en otra sección) y no las respuestas de otros.**
   * **Después: Si se le proporciona acceso a la pestaña "Respuestas Estudiantes" (según las políticas internas de la plataforma), puede ver las respuestas dadas por otros estudiantes. Esto puede facilitar el aprendizaje colaborativo, aunque se debe tener cuidado con las políticas de privacidad.**
3. **Otros Roles (Administradores, Analistas, Tutores, etc.):**
   * **Con la nueva modificación, cualquier rol con acceso a la sección de "Respuestas Estudiantes" puede analizar el rendimiento de los alumnos en actividades como Quizzes, Surveys o Exámenes de Preguntas Abiertas, sin tener que ser el creador del Learning Path.**

**Dominio del Problema**

**El dominio abarca la gestión de Learning Paths (rutas formativas) compuestos por actividades de diversa índole (cuestionarios, encuestas, recursos, exámenes abiertos). El sistema debe permitir a los usuarios:**

* **Crear y editar Learning Paths.**
* **Inscribir estudiantes en dichos Learning Paths.**
* **Registrar el progreso y las respuestas de los estudiantes.**
* **Analizar el desempeño y las respuestas de los alumnos.**

**La modificación aborda la necesidad de que la vista de las respuestas no dependa exclusivamente del rol del usuario (por ejemplo, profesor), ampliando el público que puede hacer uso de esta funcionalidad y brindando mayor flexibilidad.**

**Restricciones**

* **Seguridad y Privacidad: Aunque técnicamente se ha habilitado la visibilidad para todos los usuarios con acceso a la interfaz, idealmente se deben establecer controles de acceso. La implementación actual asume que el usuario ya tiene permiso para ver la sección. En entornos reales, puede ser necesario mantener permisos y roles para evitar filtraciones de información.**
* **Integridad de Datos: La información sobre el progreso y las respuestas de los estudiantes se carga desde archivos o una base de datos. La integridad de estos datos es esencial. No deben alterarse ni corromperse durante los procesos de lectura y escritura.**
* **Consistencia del Usuario Actual: Tras los cambios, no se ha alterado la lógica de reloadAllData(), pero se debe garantizar que los objetos User y LearningPath sigan siendo consistentes con los datos en disco.**

**Requerimientos Funcionales**

1. **Visualización de las Respuestas por Cualquier Usuario:**
   * **El sistema debe permitir que cualquier usuario con acceso a la pestaña "Respuestas Estudiantes" seleccione un Learning Path y vea la lista de estudiantes inscritos.**
   * **Al seleccionar un estudiante, el usuario puede ver las respuestas suministradas en cada actividad, independientemente de su rol.**
2. **Listado de Todos los Learning Paths:**
   * **En la pestaña de "Respuestas Estudiantes", se deben mostrar todos los Learning Paths disponibles en el sistema, sin filtrar por el creador.**
3. **Recarga de Datos:**
   * **El usuario debe poder presionar un botón "Recargar" para volver a cargar todos los datos (Learning Paths, Progresos, Usuarios) y refrescar la vista.**

**Requerimientos No Funcionales**

* **Usabilidad:  
  La interfaz debe ser intuitiva, permitiendo que cualquier usuario, incluso sin rol docente, pueda navegar por las respuestas. Deben mantenerse los controles gráficos (combo boxes, listas y áreas de texto) claros y fáciles de entender.**
* **Mantenibilidad:  
  El código debe estar estructurado de forma que sea fácil agregar nuevas funcionalidades o cambiar la lógica de filtrado sin requerir reescrituras complejas. Al haber eliminado las dependencias del rol en la lógica de visualización, se ha simplificado el mantenimiento.**
* **Escalabilidad:  
  El sistema debe funcionar eficientemente incluso si el número de Learning Paths o de estudiantes crece. El mecanismo para filtrar y mostrar datos debe ser eficiente, evitando bloqueos o demoras excesivas.**
* **Seguridad (en el contexto actual):  
  Si bien se ha eliminado la restricción por rol, es recomendable en un escenario real implementar niveles de permisos y autenticación que limiten quién puede ver qué información. Este aspecto no se aborda en la modificación actual.**

**Documentación**

**La documentación del proyecto debe reflejar los siguientes puntos:**

* **Manual del Desarrollador: Indicar cómo se cambió la lógica de filtrado de Learning Paths y la visibilidad de las respuestas. Se debe describir la nueva ubicación del código relevante (ej. createStudentResponsesPanel() en TeacherPanel).**
* **Manual de Usuario: Explicar que, a partir de esta versión, cualquier usuario con acceso a la pestaña de "Respuestas Estudiantes" podrá visualizar la información. El manual debe incluir capturas de pantalla y una guía paso a paso.**

**Rendimiento**

* **Carga de Datos:  
  El rendimiento depende de la eficiencia de la función reloadAllData() y de cómo se gestionan las colecciones internas.**
  + **Si el número de Learning Paths o estudiantes es muy grande, la operación de refresco podría tardar más.**
  + **La eliminación del filtro por rol no introduce una operación más costosa, simplemente muestra todos los datos.**
* **Presentación de Datos:  
  La interfaz gráfica opera con JTable y JList. Estas estructuras son eficientes para conjuntos de datos moderados. Para grandes cantidades de datos, podría ser necesario paginar o usar técnicas más avanzadas de virtualización.**

**Compatibilidad**

* **Compatibilidad con Diferentes Plataformas:  
  Dado que la aplicación está escrita en Java y emplea Swing, es multiplataforma por naturaleza. Este cambio no afecta la compatibilidad con diferentes sistemas operativos (Windows, Linux, macOS).**
* **Compatibilidad con Otras Partes del Sistema:  
  El cambio es interno en la lógica de visualización. Mientras el resto del código y las clases mencionadas (User, LearningPath, Progress, etc.) permanezcan compatibles, no debería haber problemas de integración.**

**Portabilidad**

* **Portabilidad del Código:  
  El código mantiene el uso de Java estándar. No se han introducido dependencias particulares. Esto facilita la portabilidad del sistema a diferentes entornos de ejecución Java.**
* **Dependencias Externas:  
  No se han agregado nuevas librerías externas. Por lo tanto, la portabilidad no se ve afectada.**

**Conclusión**

**La modificación realizada permite ampliar el alcance de la funcionalidad original, haciendo que cualquier usuario con acceso pueda visualizar las respuestas de los estudiantes. Este cambio implica un ajuste en los requerimientos, roles y restricciones. Si bien esto aumenta la visibilidad y transparencia del sistema, es necesario considerar aspectos de seguridad y privacidad según las políticas de la institución o el entorno en el que se despliegue el software. La documentación, requerimientos y consideraciones de rendimiento, compatibilidad y portabilidad se mantienen coherentes con las prácticas estándar del desarrollo en Java/Swing.**

1. **Usuarios y Roles:  
   El sistema distingue entre diferentes tipos de usuarios (estudiantes y profesores), los cuales heredan de una clase base User. Cada User tiene un role (por ejemplo, STUDENT o TEACHER), un username y una password, y métodos para autenticarse en el sistema.**
2. **Learning Paths y Actividades:  
   La clase LearningPath representa un curso o secuencia de actividades. Cada LearningPath contiene una lista de Activity (actividades), las cuales pueden ser de distintos tipos: Quiz, Survey, OpenEndedExam, ResourceReview, etc.  
   Estas actividades definen la interacción que el estudiante tendrá, ya sea responder preguntas, dar su opinión en encuestas, resolver exámenes abiertos o revisar recursos educativos.**
3. **Progreso del Estudiante:  
   La clase Progress asocia a un Student con un LearningPath y guarda información sobre el estado de cada actividad (pendiente, completada, entregada), así como las respuestas dadas a quizzes, encuestas y exámenes. Gracias a esto, el sistema puede mostrar el avance del estudiante y permitir la evaluación.**
4. **Persistencia de Datos:  
   La clase DataManager se encarga de la carga y el guardado de datos (usuarios, Learning Paths, progresos) desde y hacia el almacenamiento, manteniendo la persistencia de la información entre sesiones.**
5. **Interfaz Gráfica (GUI):  
   Las clases LoginPanel, TeacherPanel y StudentPanel representan componentes de la interfaz de usuario (usando Swing). Estas permiten:**
   * **LoginPanel: Autenticar a un usuario.**
   * **TeacherPanel: Creación de nuevas actividades, visualización de respuestas de estudiantes a actividades, y manejo de sus propios Learning Paths (si es un profesor).**
   * **StudentPanel: Permite a los estudiantes inscribirse en Learning Paths disponibles, ver sus rutas aprendidas, su progreso y las actividades que pueden realizar.**
6. **MainFrame:  
   Es la ventana principal del sistema, que contiene las distintas vistas (paneles). A través de MainFrame se autentica el usuario, se accede a los datos, y se despliegan el TeacherPanel o StudentPanel dependiendo del rol del usuario. También puede recargar datos, guardar cambios y cerrar sesión.**

**Funcionamiento general del código:**

* **Al iniciar, MainFrame carga los datos existentes (usuarios, Learning Paths, Progresos).**
* **El usuario ingresa credenciales en LoginPanel. Si la autenticación es correcta, MainFrame identifica el rol y muestra el TeacherPanel (si es profesor) o el StudentPanel (si es estudiante), o la sección que corresponda.**
* **En el TeacherPanel, el profesor (u otro usuario con acceso) puede crear nuevas actividades, ver las respuestas de los estudiantes sin restricción. También puede consultar estadísticas anuales.**
* **En el StudentPanel, el estudiante ve en una pestaña sus Learning Paths y el progreso de las actividades. En otra pestaña, ve qué Learning Paths están disponibles para inscribirse. Al seleccionar una actividad, el estudiante puede interactuar con ella (responder un quiz, completar un examen abierto, etc.).**
* **Los cambios (nuevos usuarios, respuestas, progreso) se guardan a través de DataManager.**