

Especificación de implementación: flujo post /study-planner/create (Tracking Learn + Cierre por inactividad + Cumplimiento en Dashboard)

Contexto y objetivo

Una vez que el usuario crea su plan/sesiones en **/study-planner/create**, al estudiar dentro de **Learn** el sistema debe:

1. Detectar **inicio real** de una lección/sesión (no al entrar a la vista, sino cuando realmente empieza).
2. Detectar **fin de la lección** y, cuando aplique, cerrar la **study_session**.
3. Evitar depender de botones o timers del navegador (porque el usuario puede cerrar pestaña/navegador).
4. Implementar un **cron real por inactividad** (server-side), con análisis dinámico.
5. En **/study-planner/dashboard**, calcular y reflejar **cumplió / no cumplió** (por día), y ejecutar el reconcile/reprogramación sumándolo a lo que ya hace.

Nota: La **pantalla/modal de descanso** se deja **para el final** (fuera de este alcance) por falta de material/definición de UX.

0) Logout automático por inactividad (solo en /courses/[slug]/learn)

Objetivo

Implementar un **logout automático por inactividad únicamente** dentro de la ruta **/courses/[slug]/learn**, utilizando el sistema de ayuda automatizada de **LIA** y la detección de inactividad soportada por **RRWEB**.

Alcance

- El logout automático aplica **exclusivamente** en **/courses/[slug]/learn**.
- No debe afectar el resto de rutas de la plataforma (dashboard, study-planner, course overview, etc.).
- La inactividad se detecta con los eventos/captura de **RRWEB** integrados al sistema de LIA, evitando depender únicamente de timers del navegador.
- El criterio de inactividad será: **60 minutos continuos sin actividad** dentro de la ruta **/courses/[slug]/learn**.

Comportamiento esperado

- Si el usuario permanece **60 minutos inactivo** dentro de /courses/[slug]/learn:
 - Se ejecuta el **logout** de la plataforma.
 - Se redirige al usuario a la pantalla de login (o ruta definida por el flujo actual de autenticación).
- Si el usuario retoma actividad antes de los 60 minutos:
 - Se reinicia el conteo de inactividad (y no se dispara el logout).

Consideraciones técnicas

- Debe registrarse un “último evento de actividad” (ej. interacción UI / eventos RRWEB) para evaluar la inactividad real.
- Idealmente el logout debe ser **idempotente** (si ya está deslogueado o la sesión expiró, no romper el flujo).

Estimación

- **Tiempo estimado de desarrollo: 2 horas.**

1) Definiciones

- **Study Session (study_sessions):** entidad que agrupa una sesión de estudio planificada y su ejecución real.
 - **Lesson Tracking (nuevo o equivalente):** registro/estado para saber qué lección se está ejecutando, con timestamps para video, materiales, LIA y actividad.
 - **Estimaciones de tiempo:**
 - **T_lección** = tiempo estimado total de la lección
 - **T_video** = duración del video
 - **T_materiales** = suma de tiempos estimados de lecturas/materiales/etc.
 - **T_restante** = $T_{lección} - T_{video} - T_{materiales}$ (clamp mínimo a 0)
-

2) Inicio de sesión / inicio de lección (Start)

Regla: La sesión/lección se considera iniciada cuando el usuario **reproduce el video** (primer play).

Evento sugerido: video_play

Acción: si el tracking de esa lección no está iniciado:

- Guardar started_at = now()
 - Marcar status = in_progress (si aplica)
 - Guardar start_trigger = "video_play"
-

3) Fin de lección / fin de sesión: 3 flujos

Flujo A — Lección con Quiz

Regla: Si hay quiz, la lección (y la sesión, si aplica) termina cuando el usuario “termina el quiz”.

Acción al evento final:

- completed_at = now()
 - status = completed
 - end_trigger = "quiz_submitted"
 - Este flujo no depende del cron por inactividad.
-

Flujo B — Lección con actividad obligatoria con LIA (cron por inactividad + análisis dinámico)

B.1 Regla de análisis dinámico (primer análisis)

El análisis se programa con base en:

$T_{\text{restante}} = T_{\text{lección}} - T_{\text{video}} - T_{\text{materiales}}$

Muy importante: el tiempo de espera **empieza a contar** en el momento en que el usuario **interactúa con LIA por primera vez** dentro de esa lección.

Evento clave: lia_first_message_at (primer mensaje del usuario a LIA)

Acción inmediata al primer mensaje:

- Guardar `lia_first_message_at = now()`
- Guardar `lia_last_message_at = now()` (y actualizarlo en cada mensaje)
- Calcular y guardar `next_analysis_at = lia_first_message_at + T_restante`

B.2 Cron recurrente (cada 5 minutos)

Después de ejecutar el primer análisis, el cron debe ejecutarse **cada 5 minutos**:

- `next_analysis_at = now() + 5min`

El cron es **server-side** y no depende de que el usuario mantenga abierta la pestaña.

B.3 Condición de completado por inactividad

Regla: se marca la lección como completada cuando:

- Han pasado **5 minutos** desde el último mensaje enviado por el usuario a LIA
→ `now - lia_last_message_at >= 5min`

Acción al cumplir condición:

- Marcar lección como completada:
 - `lesson_tracking.completed_at = lia_last_message_at + 5min` (o `now()`, pero preferible lo primero)
 - `lesson_tracking.status = completed`
 - `completion_reason/end_trigger = "lia_inactivity_5m"`

B.4 Cierre inmediato por cambio de contexto (salida de la lección)

Además del cron, se debe **finalizar** la lección (y potencialmente la sesión) si el usuario:

- Avanza a la **siguiente lección**, o
- Realiza cualquier acción dentro de la plataforma que implique que **ya no está haciendo esa lección** (navega fuera, abre otro módulo, inicia otra lección, etc.)

Regla práctica: si llega un evento de frontend que indique “cambio de lección / cambio de contexto” y ya no coincide el `lesson_id`, entonces:

- cerrar el tracking de la lección actual con `end_trigger = "context_changed"`

- y evaluar si con eso ya debe cerrarse la study_session (si era la última parte pendiente de la sesión)
-

Flujo C — Lección sin Quiz y sin LIA (aplica “lo mismo” que B)

Se aplicará la **misma lógica general**: análisis dinámico + cron por inactividad, pero usando actividad general en lugar de mensajes.

C.1 Primer análisis dinámico

Calcular:

- $T_{\text{restante}} = T_{\text{lección}} - T_{\text{video}} - T_{\text{materiales}}$

Como no existe lia_first_message_at, se define un evento equivalente:

Evento sugerido: post_content_start_at (cuando ya terminó/consumió lo esencial del contenido)

Ejemplos de disparadores medibles:

- terminó video (video_ended_at) y/o
- abrió/consumió el material final requerido y/o
- llegó al estado UI “fin de lección” (aparece CTA para continuar)

Acción:

- Guardar post_content_start_at = now()
- Programar next_analysis_at = post_content_start_at + T_restante

C.2 Cron recurrente (cada 5 minutos)

- Luego del primer análisis: next_analysis_at = now() + 5min

C.3 Condición de completado por inactividad

Como no hay chat, usar:

- last_activity_at (se actualiza con eventos relevantes dentro de la lección: scroll/abrir material/video events/heartbeat liviano)

Regla: completar cuando:

- $\text{now} - \text{last_activity_at} \geq 5\text{min}$

Acción:

- `completed_at = last_activity_at + 5min` (o `now()`)
- `end_trigger = "activity_inactivity_5m"`

C.4 Cierre inmediato por cambio de contexto

Igual que flujo B: si avanza a siguiente lección o sale del contexto de esa lección → cerrar tracking con `end_trigger = "context_changed"` y evaluar cierre de `study_session`.

4) Cron real + casos extremos (cerró ventana / no volvió en días)

Requisito

El cierre por inactividad **no puede depender** del frontend.

Implementación requerida

1. Persistir en BD los timestamps mínimos:
 - B: `lia_first_message_at`, `lia_last_message_at`, `next_analysis_at`
 - C: `post_content_start_at`, `last_activity_at`, `next_analysis_at`
 2. Un **Scheduled Job server-side** que corre cada 5 minutos:
 - busca trackings activos con `next_analysis_at <= now()`
 - ejecuta reglas de B o C
 - si completa → marca `completed_at/status/reason`
 - si no completa → actualiza `next_analysis_at = now() + 5min`
 3. **Fallback / Catch-up (recomendado y necesario):**
 - En requests relevantes (dashboard, entrar a learn, evento LIA, etc.) ejecutar un reconcile liviano:
 - si hay trackings activos vencidos (`next_analysis_at` en pasado), procesarlos también ahí.
 - Esto cubre escenarios donde el cron falló temporalmente o hubo downtime.
-

5) Cierre de Study Session (cuando termina la lección)

Cuando una lección se complete por cualquier flujo:

- Evaluar si esa lección era la última pendiente dentro de la sesión.
- Si ya corresponde, marcar `study_session` como completada:
 - `setear study_sessions.completed_at`
 - `setear actual_duration_minutes` con base en `started_at` vs `completed_at`
 - `status = completed`

No depender de un botón; el cierre debe ocurrir por eventos y cron.

6) Cumplió / no cumplió en `/study-planner/dashboard` (sumado a lo que ya hace)

Regla de negocio principal

- Importa si el usuario **cumplió las sesiones el día que le correspondía**.
- La **hora** exacta no es el criterio principal.
- Si la sesión se completó en otro horario pero el mismo día → cuenta como **cumplió** (no reprogramar).
- Si el día pasó y la sesión no se completó → **no cumplió** y se activa el flujo de missed/reprogramación.

Dónde se ejecuta

Al cargar `/study-planner/dashboard`:

1. Ejecutar reconcile de sesiones vencidas (las que ya pasaron de día y no están completadas) y aplicar “missed + reprogramación” según la lógica actual del sistema.
2. Calcular/mostrar el estado de “cumplió / no cumplió” por día como parte de stats.

Validación requerida

Para poder cerrar este punto (6), necesito validación de Ernesto de que **lo que hoy hace dashboard** (missed/reprogramación/stats) **sí corresponde** con lo que él quiere.

Estimaciones y dependencias

Punto 5 (cron real + tracking Learn: flujos B y C)

- **Estimación mía como desarrollador: 2 días** para terminar hasta el **punto 5 completo** (incluye flujos B y C con cron por inactividad, triggers de “salíó de la lección, persistencia mínima y pruebas básicas, esto en caso de que el cron no implique mucho problema al desarrollarse ya que esto es algo nuevo para mi pero necesario para que se haga bien lo que se quiere implementar
- **Inicio del conteo:** este estimado cuenta **a partir del momento en que se aprueben por escrito estas tareas/especificaciones.**

Punto 6 (cumplió/no cumplió en dashboard)

Requiero validación de Ernesto sobre el dashboard actual.

- **Caso A — Si el dashboard actual sí funciona como Ernesto quiere:**
 - **Estimación adicional: ~0.5 día** (medio día) para implementar el punto 6 y probarlo. La aprobación de esto sugiere que ya se probó exhaustivamente por Ernesto que lo implementado en en study-planner/dashboard
- **Caso B — Si el dashboard actual no coincide con lo que Ernesto quiere:**
 - Se requiere **revisar y planear** el comportamiento final (como ahorita con el punto 5) antes de terminar el punto 6.
 - En este caso no puedo comprometer el “medio día” hasta que se definan reglas finales, correcciones, mejoras y alcance.
- La aceptación de este documento implica que **ya se validó y probó exhaustivamente** que todas las dependencias y funcionalidades necesarias para ejecutar lo aquí descrito **operan correctamente en el ambiente actual**, incluyendo el estado posterior al **cambio de dominio** y a los **cambios extensivos** aplicados al sitio en las últimas semanas. En caso de detectarse regresiones, fallas o comportamientos inesperados derivados de dichos cambios, **los tiempos estimados comenzarán a contar únicamente a partir de que la plataforma sea confirmada como estable** y las dependencias críticas estén funcionando correctamente. Como referencia, actualmente se identificó que en /learn el asistente **LIA está respondiendo en 2 idiomas** y además **ya no está disponible el botón para cambiar el idioma**, por lo que estos puntos deben corregirse y/o confirmarse antes de considerar “aprobado” el inicio de los trabajos descritos en este documento.
- Adicionalmente, cualquier **modificación, ajuste o requerimiento nuevo** solicitado durante el desarrollo de lo planteado en este documento será **redactado por escrito**

y enviado para **aprobación explícita** antes de implementarse. Dichos cambios pueden **impactar el plazo de entrega originalmente estimado**, dependiendo de su complejidad y de la etapa de implementación en la que se soliciten. Para cada cambio solicitado se generará un **documento/actualización de alcance** que incluirá la **estimación revisada** (probable aumento o reducción del tiempo de entrega), calculada tras revisar técnicamente lo solicitado.