

3w - Definitivo - Personalización #1325

Optimización crítica de consulta que lista materias

2026-06-11 02:14 - Fernando Martinez

Estado:	Cerrada	Fecha de inicio:	2026-06-10
Prioridad:	Alta	Fecha fin:	
Asignado a:	Federico Milicich	% Realizado:	100%
Categoría:		Tiempo estimado:	0.00 hora
Versión prevista:	3.19.2.1	Tiempo dedicado:	0.00 hora

Descripción

Optimización de Consulta de Materias Vigentes (Módulo Inscripciones)

1. Contexto y Análisis del Cuello de Botella

Durante la auditoría de rendimiento del módulo de autogestión (3w), se detectó que la consulta encargada de listar las actividades/materias disponibles para la inscripción de un alumno presentaba altos tiempos de respuesta (promedio original de ~644ms), lo que provocaba saturación de CPU en la base de datos durante los picos de inscripción masiva.

El análisis del plan de ejecución (EXPLAIN ANALYZE) arrojó los siguientes problemas arquitectónicos:

- **Carga procedural excesiva:** El uso de la función de base de datos `f_actividades_con_comisiones_vigentes` consumía el 98% del tiempo total de la consulta (631ms) debido al procesamiento fila por fila interno de PL/pgSQL.
- **Escaneos históricos completos:** Las cláusulas `NOT IN` (`SELECT * FROM vw_hist_academica_basica`) forzaban al motor a recuperar todo el historial de exámenes y equivalencias del alumno para armar tablas hash temporales en memoria, en lugar de cortar el escaneo temprano.
- **Subconsultas por fila en SELECT:** El nombre de la responsable académica se evaluaba con un (`SELECT ... LIMIT 1`) anidado dentro de la columna del `SELECT` principal.

2. Solución Estructurada

Para reducir drásticamente los tiempos de carga, se reestructuró la consulta combinando inlining relacional, empuje de predicados (predicate pushdown) y creación de índices.

A. Creación de Índice en Base de Datos

Se comprobó que PostgreSQL realizaba un Seq Scan sobre más de 146.000 registros en la tabla `sga_comisiones`. Para mitigar esto, es obligatorio crear el siguiente índice compuesto filtrado:

```
CREATE INDEX idx_sga_comisiones_insc_activa
ON sga_comisiones (periodo_lectivo, estado, inscripcion_habilitada)
WHERE inscripcion_habilitada = 'S' AND estado IN ('A', 'P');
```

B. Refactorización en la Clase PHP (lista_materias)

Se personalizó la función `lista_materias($parametros)` implementando 3 mejoras directas sobre el armado dinámico del SQL:

1. **Inlining de la Función:** Se desarmó `f_actividades_con_comisiones_vigentes` extrayendo únicamente la lógica necesaria de filtros de comisiones (fechas, interfaz, modalidad y ubicación). Se integró como un subquery puro en el bloque de `JOIN`, permitiendo que el planificador de Postgres resuelva el cruce usando índices nativos de una sola vez. Se usó `SELECT DISTINCT` para replicar la deduplicación del `IN` original.
2. **Reemplazo de IN por EXISTS:** Se cambiaron todas las validaciones históricas y de regularidad a `NOT EXISTS`. Se cambió también el bloque PHP de `select_regularizada` para que evalúe la variable `regularizada` usando un `EXISTS` en lugar de `COUNT(1) > 0`.
3. **Migración a LATERAL JOIN:** La obtención de la responsable académica se movió del `SELECT` a un `LEFT JOIN LATERAL`.

3. Resultados y Métricas de Impacto

Tras la aplicación del índice y la refactorización PHP, se realizó una comparación de resultados lógicos demostrando que la nueva consulta devuelve una coincidencia exacta de datos (0 diferencias).

- **Tiempo Inicial Promedio:** ~644 ms
- **Tiempo Final Promedio:** 138 ms
- **Reducción de Tiempo:78.5% más rápido** (Ahorro de ~506ms por ejecución).

Esta ganancia es crítica para la concurrencia: En un pico de 1.000 alumnos concurrentes, el tiempo acumulado de procesamiento en la base de datos se redujo de **~11 minutos a poco más de 2 minutos**, liberando enormemente los núcleos del servidor PostgreSQL.

Revisiones asociadas

Revisión 101367 - 2026-06-11 02:14 - Fernando Martinez

Ticket #1325

Histórico

#1 - 2026-06-11 02:16 - Fernando Martinez

Se deja creado el índice en -test y producción.
Se aplica la nueva función en -test.
Se aguarda confirmación para aplicar en producción.

#2 - 2026-06-11 02:24 - Fernando Martinez

- Estado cambiado de *En desarrollo* a *Listo para testeo*
- Asignado a cambiado de *Fernando Martinez* a *Federico Milicich*
- % Realizado cambiado de *0* a *90*

#3 - 2026-06-11 13:50 - Federico Milicich

- Estado cambiado de *Listo para testeo* a *En testeo*

#4 - 2026-06-11 23:23 - Federico Milicich

- Estado cambiado de *En testeo* a *Cerrada*
- % Realizado cambiado de *90* a *100*

Verificado.

#5 - 2026-06-12 14:33 - Fernando Martinez

Se revierte la personalización y se deja la función original del SIU, ya que la evidencia en tiempo real en la inscripción demostró que la verdadera ganancia la dio la creación del índice.
Como la ganancia evidenciada entre funciones es marginal (30-50ms), se prefiere dejar la lógica estándar del SIU.