Sí, al ejecutar:

$data = Excel::toArray([], $rutaArchivoExcel)[0];

dd($data);

Estás cargando **todo** el contenido del archivo Excel en memoria antes de procesarlo. Esto puede afectar a Laravel si el archivo es muy grande, porque podrías consumir **mucha memoria RAM** y ralentizar o incluso hacer que la aplicación falle por falta de memoria.

**🚀 Soluciones para optimizar la lectura del archivo**

Si el archivo Excel es grande, en lugar de toArray(), usa **un importador basado en filas** con Laravel Excel, que procesará los datos en **chunks** (fragmentos), evitando consumir demasiada memoria.

**Opción 1: Usar LazyCollection (Eficiente en memoria)**

use Maatwebsite\Excel\Facades\Excel;

$data = Excel::toCollection([], $rutaArchivoExcel)->first();

foreach ($data as $row) {

// Procesar cada fila aquí sin cargar todo en memoria

dd($row); // Ver la estructura de una fila

}

Esto carga el archivo fila por fila y no lo mantiene todo en memoria.

**Opción 2: Usar chunk() (Procesa por partes grandes)**

Si necesitas procesar muchas filas, usa chunk() para procesarlas en bloques:

use Maatwebsite\Excel\Facades\Excel;

Excel::import(new NotificacionesHaciendaImport, $rutaArchivoExcel)->chunk(100);

Esto cargará el archivo en partes de 100 filas a la vez.

**¿Cuándo usar toArray()?**

toArray() es útil solo si el archivo es **pequeño** (menos de 1000 filas). Si esperas archivos grandes, usa LazyCollection o chunk().

Con estos cambios, evitarás problemas de rendimiento y errores en Laravel. 🚀