**Decisiones adquiridas durante el desarrollo de la prueba:**

**Respetar la estructura de datos proporcionada:** Personalmente hubiera embebido las imágenes de los anuncios en cada anuncio solo con el campo “url” y “quality”. Creo que en un futuro la edición de las imágenes se haría por cada anuncio y no penalizaría tanto tener algo de información repetida como un anuncio que comparta imagen con otro como la penalización que se tiene en colecciones separadas. Para devolverlas embebidas en el anuncio se ha optado por hacer un “lookup” de mongo para aprovechar el motor de base de datos. El problema es un entorno productivo de esta aplicación no podríamos tener colecciones compartidas entre varias instancias de mongoDB, lo que perjudicaría bastante la aplicación para una cantidad grande de anuncios.

**Paginación:** Se ha optado por un scroll infinito ya que era lo que Groummet ofrecía. Era la primera vez que usaba Groummet y no sabia que no ofrecería ningún tipo de tabla con paginación como las que he usado con Prime, pero personalmente quería probarlo y he visto esta prueba como una oportunidad para ello. La paginación se ha hecho usando “skip” y “limit” en mongo pero esta solución se vuelve muy ineficiente cuando se dispone de muchos datos con los que tratar. En un futuro esta solución se tendría que mover a usar cursores de mongoDB para afinar y mejorar el rendimiento de las consultas a mongoDB.

<https://www.codementor.io/@arpitbhayani/fast-and-efficient-pagination-in-mongodb-9095flbqr>

**Autenticación:** Con el objetivo de usar solo GraphQL para la comunicación entre todos los componentes de la aplicación he adaptado el endpoint standard “/.well-known/jwks.json” de auth0 a graphql a la query “jsonWebToken”

**Ficheros. env:** He decidido subir los ficheros .env porque no hay información sensible. En teoría estos ficheros deberían de estar ignorados y cada desarrollador tener sus propios ficheros para levantar sus entornos en local. Pero para hacer más fácil la inicialización de la aplicación se han subido a Git.

**Problemas encontrados:**

**Dockerizacion de la aplicación:** Al ser la primera vez que usaba un workspace de npm para la parte del servidor y compartir las dependencias de los micro servicios, he tenido problemas a la hora de construir las imágenes de docker, por esa razón he tenido que duplicar las dependencias entre los micro servicios. Toda la información encontrada por internet no me ha parecido de una solución idónea al problema. Las dependencias de desarrollo las he podido dejar en el package.json padre porque en local si se instalan satisfactoriamente.

**A tener en cuenta:**

**Refrescar después de editar**: Por falta de tiempo, después de editar una descripción de un anuncio debéis refrescar la pagina para ver los cambios afectados.

**Calculo de la puntuación:** En los scripts de migración observareis que las puntuaciones no estén correctas, esto es debido, a que se hizo un calculo inicial y se crearon los scripts con esa exportación de la base de datos, pero el calculo se ha mejorado contando las palabras mejor y evitando los signos de puntuación que pueden estar pegados a las palabras. Si editáis una descripción se recalculara la puntuación de manera adecuada.

**Grafico**: Hay un bug del grafico de Groummet que con solo un valor no puede pintar el grafico. Debéis modificar varias veces una descripción para poder visualizar el grafico correctamente.

