

RESULTADOS DE PRUEBAS DE DESEMPEÑO

Justificación

Dado que las consultas acerca de los listados (búsqueda, filtrado y ordenamiento) en el modelo de auditoría, se realizarán frecuentemente sobre los campos “author”, “action” y “model_name”, se evaluar qué ventajas o desventajas ofrece agregar índices a estos campos. No obstante, hay que tener en cuenta que la operación más frecuente a efectuar es la inserción sobre la tabla y los índices tienden a ralentizarla puesto que es necesario preservar el orden.

Diseño de la prueba

Para cargar la tabla de acciones auditables, se elaboró un script parametrizado, en Python, que genera un archivo tipo JSON que puede ser utilizado como un *Fixture*. El *script*, toma como entrada un archivo de configuración, llamado “fixture_settings.py”, en el que se debe indicar el nombre del modelo a agregar (“name”), cuantos elementos (“size”) y qué campos posee (“fields”). El *script* también crea usuarios en la base de datos para que puedan ser referenciados por el campos “author”.

En la figura 1 se muestra un ejemplo del archivo utilizado para cargar la base de datos con un millón de elementos en la tabla de acciones auditables. Se pueden incluir otras tablas si así se desea.

```
MODELS = [  
    {  
        'size': 1000000,  
        'name': 'auditor_core.AuditableAction',  
        'fields': [  
            'author',  
            'action',  
            'datetime',  
            'model_name',  
            'model_json_old',  
            'model_json_new'  
        ]  
    },  
]
```

Figura 1. Ejemplo de “fixture_settings.py”

La prueba se realizó cuatro veces, con mil, diez mil, cien mil y un millón de registros en la tabla de acciones auditables. Luego, se insertaron cien registros en dos tablas, una auditable y otra no auditable, y se comparan los tiempos obtenidos cuando la tabla posee índices y cuando no.

Este proceso se realizó con diferentes manejadores de base de datos relacionales, SQLite, MySQL y PostgreSQL para obtener el desempeño en cada uno de ellos.

Análisis de resultados

Cada uno de los resultados presentados fueron generados siguiendo lo explicado anteriormente sobre un computador con las siguientes especificaciones:

- Memoria RAM de 8GB.
- Procesador Intel® Core™ i5-4300M CPU @ 2.60GHz × 4.
- Tarjeta de vídeo Intel® Haswell Mobile.
- Sistema Operativo de 64 bits.
- Espacio en disco de 512GB.

SQLite

A continuación se presentan los resultados de las pruebas para el manejador de base de datos SQLite.

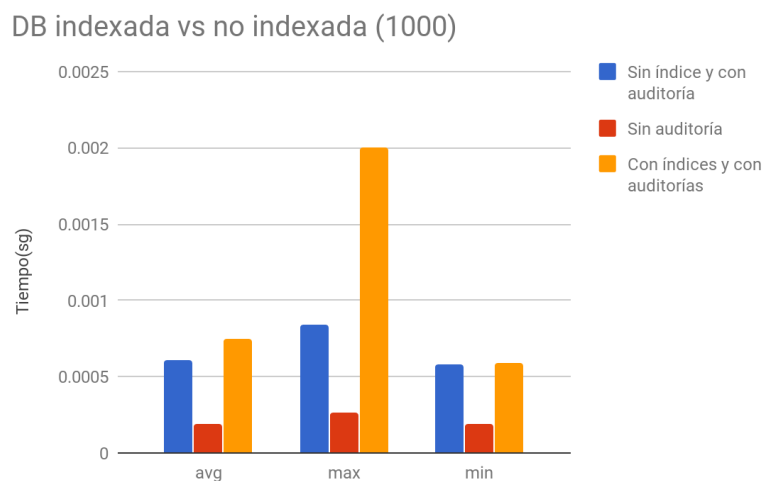


Figura 2. Resultados del promedio, mínimo y máximo para la tabla indexada comparada con la no-indexada al cargar mil registros.

El color azul representa el tiempo inserción de un registro en una tabla que es auditable, “AuditableDummyModel” sin agregar los índices. El color rojo representa el tiempo de inserción en una tabla que no es auditable, “DummyModel” y sin índices. El color amarillo representa el tiempo inserción de un registro en “AuditableDummyModel” agregando índices a la base de datos.

Con los resultados mostrados en la figura 2, se observa un incremento poco significativo entre el agregar o no índices a las tablas. En promedio, este valor es del 22%. Al comparar ambos valores, con

el tiempo de inserción en la base de datos de un registro sin auditar, se presenta un crecimiento de 212% y 283% con respecto a sus promedios.

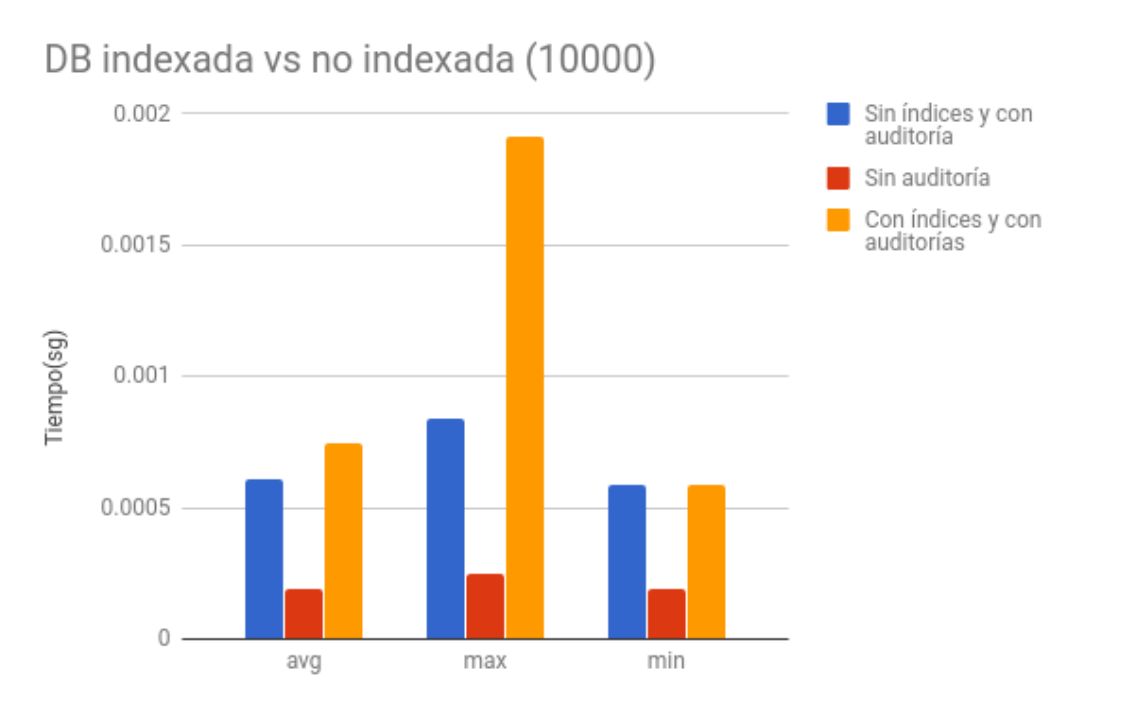


Figura 3. Resultados del promedio, mínimo y máximo para la tabla indexada comparada con la no-indexada al cargar diez mil registros.

En este caso, el la diferencia entre las tabla indexada y la que no lo estaba fue de 23% y comparados con el tiempo de inserción sin trazas de auditoría se incrementó en 213% 286% respectivamente. La tabla indexada tiene un tiempo promedio de inserción (tomando en cuenta el del registro auditable) de 0.000748589 segundos.

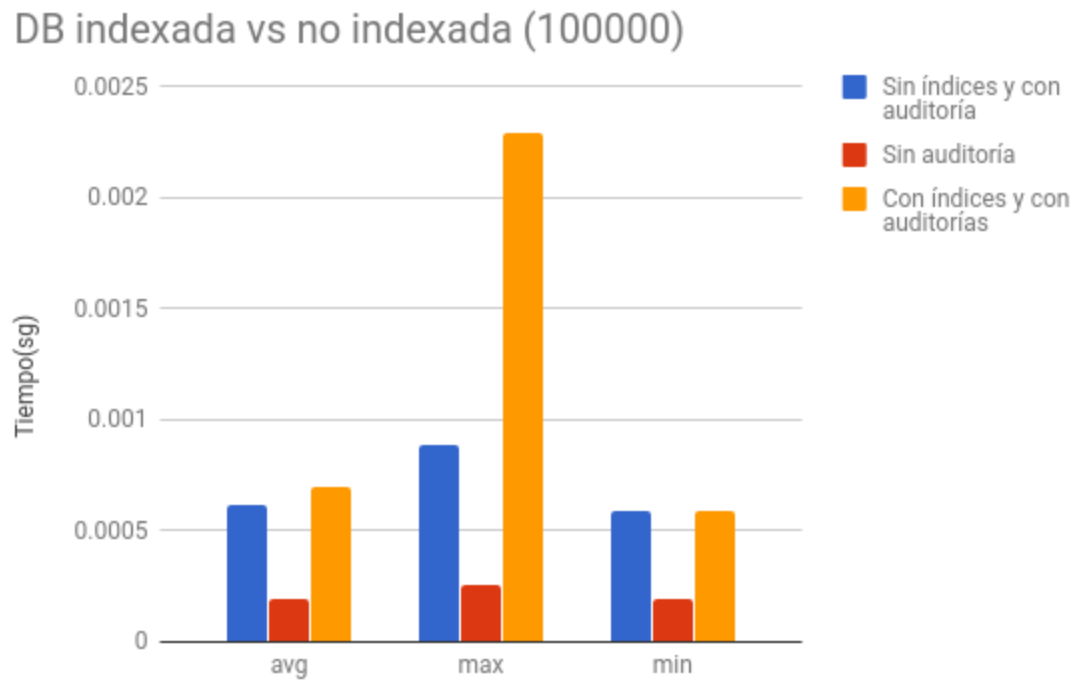


Figura 4. Resultados del promedio, mínimo y máximo para la tabla indexada comparada con la no-indexada al cargar cien mil registros.

El comportamiento presentado tanto para la inserción de mil registros, como para la de cien mil, se mantiene al aumentar la carga de la base de datos SQLite a cien mil registros.

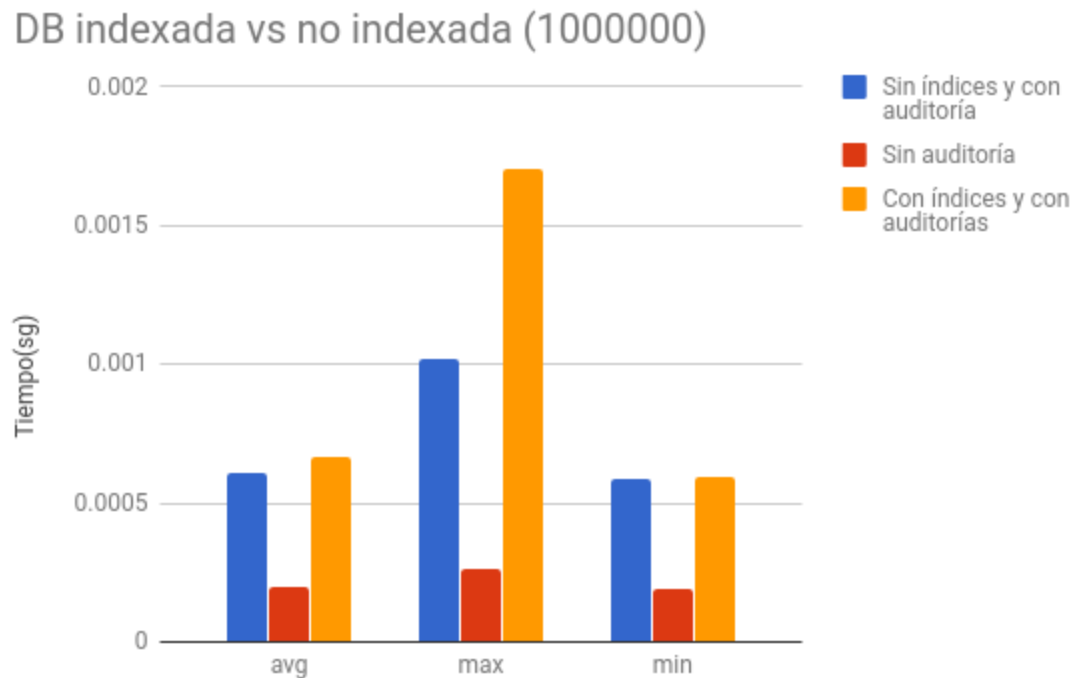


Figura 5. Resultados del promedio, mínimo y máximo para la tabla indexada comparada con la no-indexada al cargar un millón de registros.

Las inserciones en la tabla auditable con índices conllevan un mayor tiempo de ejecución que el resto, como era esperado. Considerando todos los resultados anteriores, se puede concluir que para el manejador de base de datos SQLite el tiempo de inserción de un registro auditable cuando la tabla posee índices es de 0.0007, en promedio. Este valor, representa un aumento sobre el valor sin índices de aproximadamente 17% y sobre la inserción sin auditorías de 250%.

Sin embargo, esta diferencia es prácticamente indetectable por un humano por lo que, agregar índices, es una solución viable para mejorar el tiempo de consulta en el futuro con la base de datos SQLite. Es importante resaltar que los tiempos no desmejoraron al aumentar la carga del manejador.

Tabla 1. Resumen de los promedios de crecimiento usando SQLite

Resumen		
%Crecimiento sin indices vs sin auditorías	%Crecimiento con indices vs sin auditorías	% Crecimiento sin índices vs con índices
213.2649828	267.2000846	17.21908549

PostgreSQL

Luego de efectuar las pruebas sobre SQLite, se modificó la configuración en Django, para utilizar PostgreSQL, en donde se obtuvo los resultados mostrados a continuación:

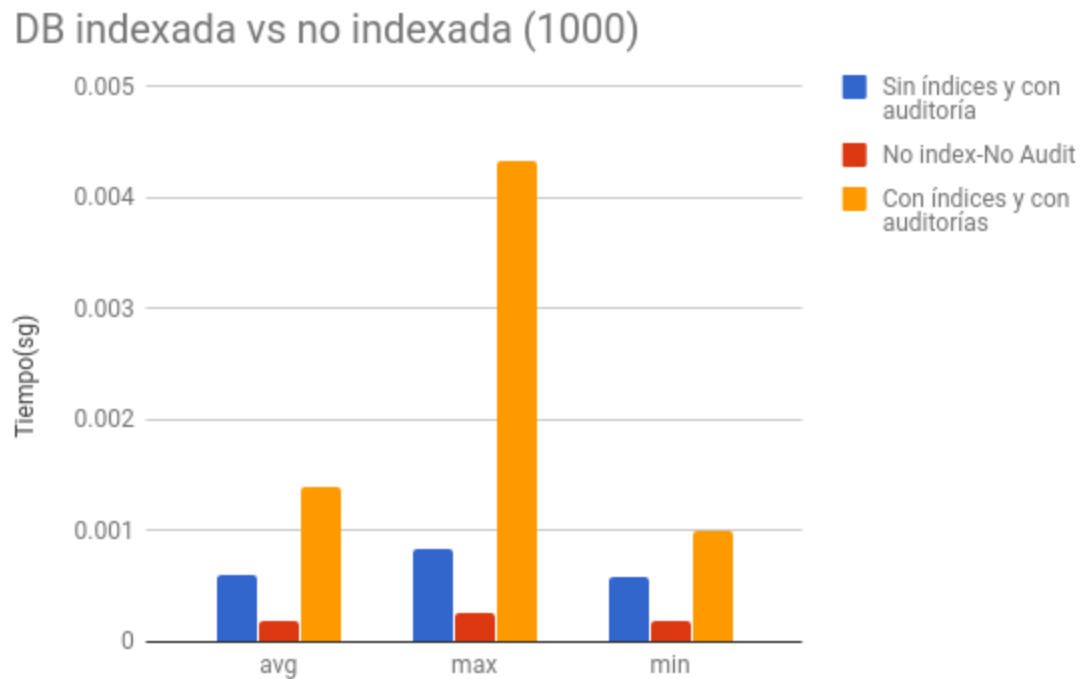


Figura 6. Resultados del promedio, mínimo y máximo para la tabla indexada comparada con la no-indexada al cargar mil registros.

En este caso también se observa que el tiempo de inserción en la tabla sin índices es menor, en un 129%. El tiempo promedio de creación de un registro sin auditorías es de 0.0001940346 segundos y con respecto a este valor, agregar índices a la tabla aumenta el tiempo en 616%.

DB indexada vs no indexada (10000)

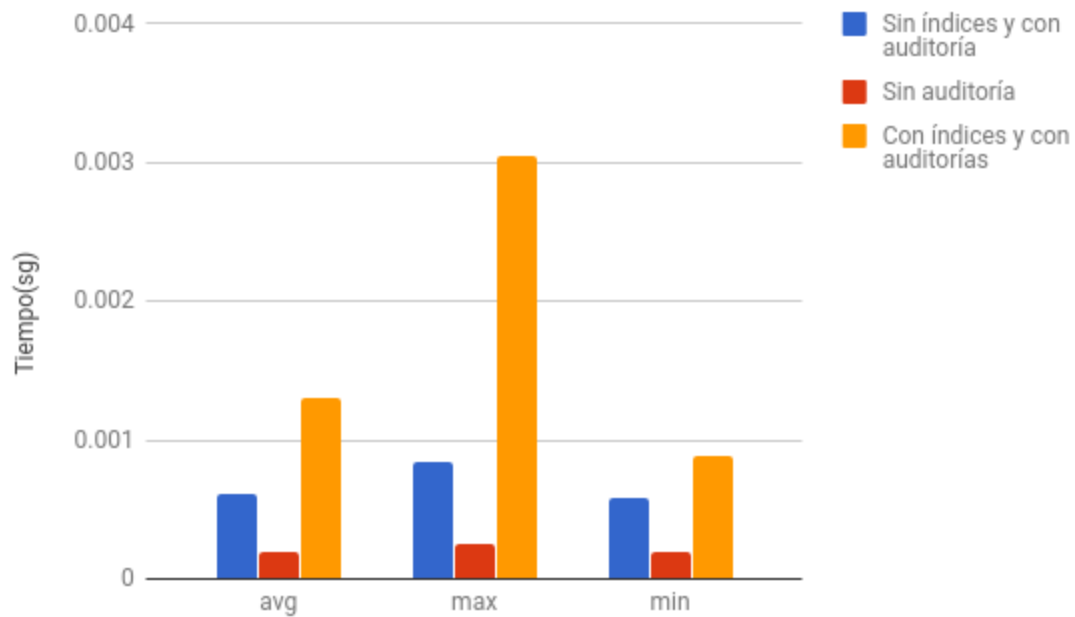


Figura 7. Resultados del promedio, mínimo y máximo para la tabla indexada comparada con la no-indexada al cargar diez mil registros.

Al insertar un registro auditable se obtiene un aumento de 213% y 569%, con y sin índices respectivamente. Entre estos valores, se tiene una diferencia de 113%, siendo el segundo mayor que el primero.

DB indexada vs no indexada (100000)

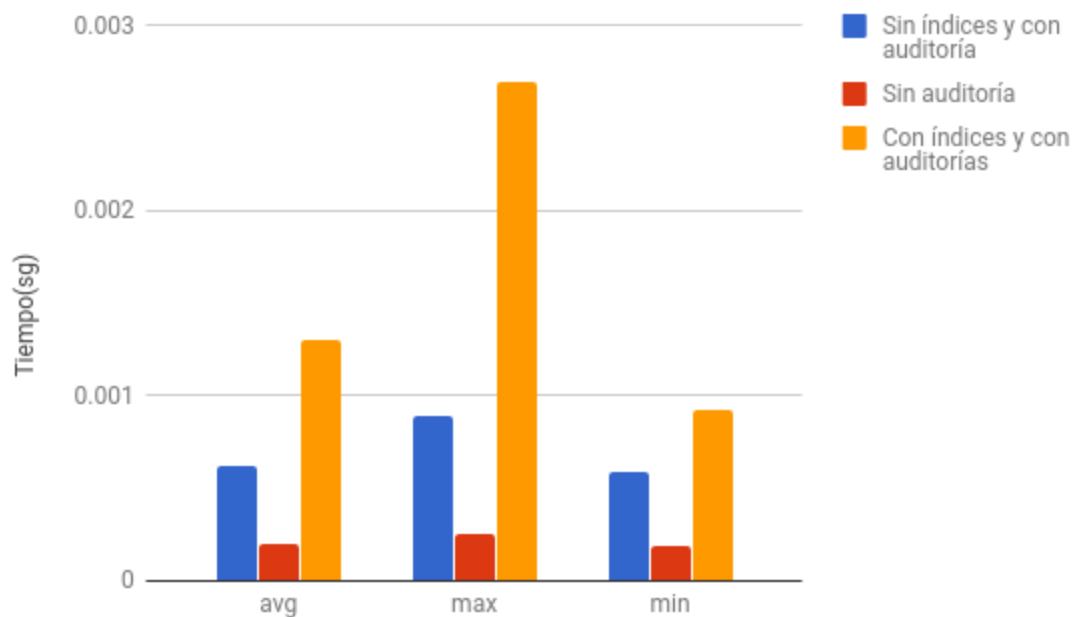


Figura 8. Resultados del promedio, mínimo y máximo para la tabla indexada comparada con la no-indexada al cargar cien mil registros.

DB indexada vs no indexada (1000000)

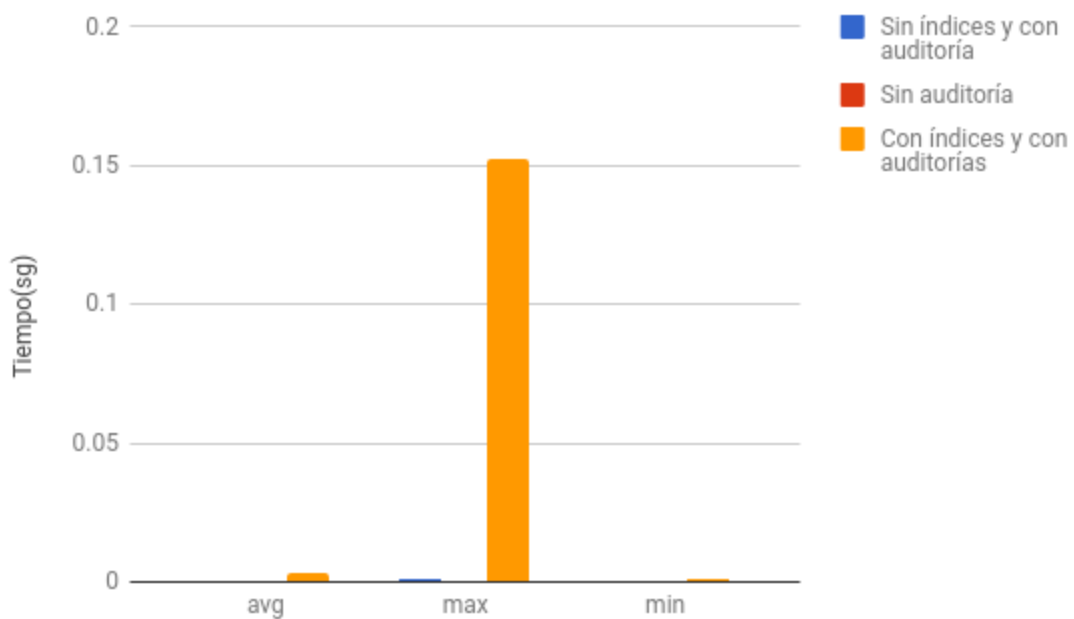


Figura 9. Resultados del promedio, mínimo y máximo para la tabla indexada comparada con la no-indexada al cargar un millón de registros.

En las figuras 8 y 9 se observa que la creación del registro con índices y con auditorías, es mayor al resto. En general, se puede concluir que para PostgreSQL se presenta un incremento promedio de 182% al comparar los valores de ambas inserciones con auditorías.

A diferencia de SQLite, PostgreSQL, muestra una pequeña desmejora en el desempeño al cargar un millón de registros.

Tabla2. Resumen de los promedios de crecimiento usando PostgreSQL

Resumen		
%Crecimiento sin indices vs sin auditorías	%Crecimiento con indices vs sin auditorías	% Crecimiento sin índices vs con índices
213.2649828	786.5247172	182.9698376

MySQL

Por último, se efectuaron las pruebas sobre MySQL, son lo que se obtuvo las gráficas mostradas a continuación.

DB indexada vs no indexada (1000)

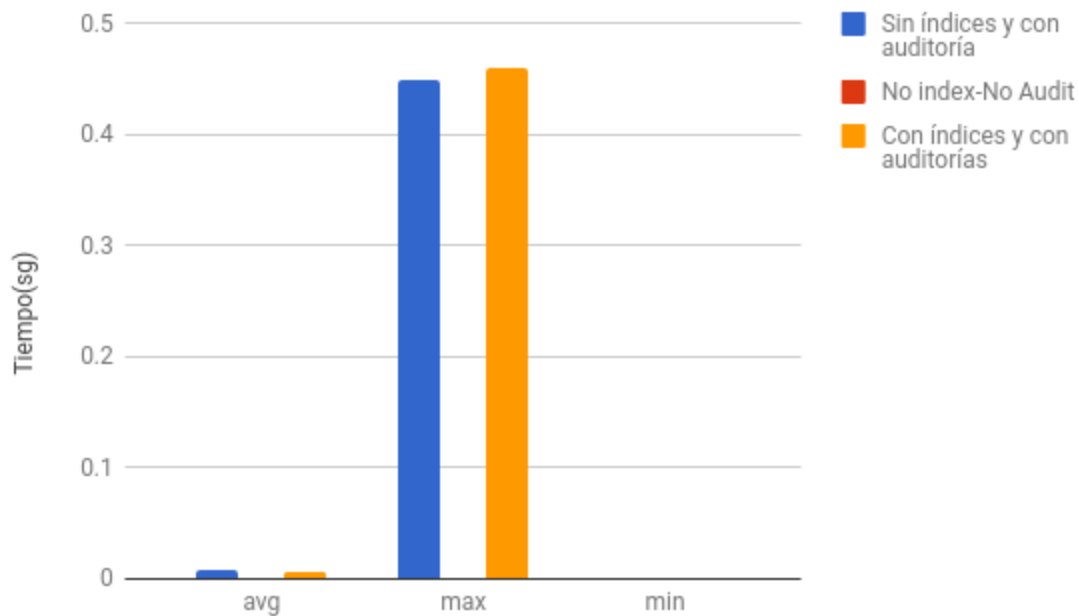


Figura 10. Resultados del promedio, mínimo y máximo para la tabla indexada comparada con la no-indexada al cargar mil registros.

Este manejador de base de datos es significativamente más lento que el resto, sin embargo el comportamiento descrito se mantiene. El uso de auditorías aumenta el tiempo de inserción en 1532% y 1751% sin índices y con índices, respectivamente.

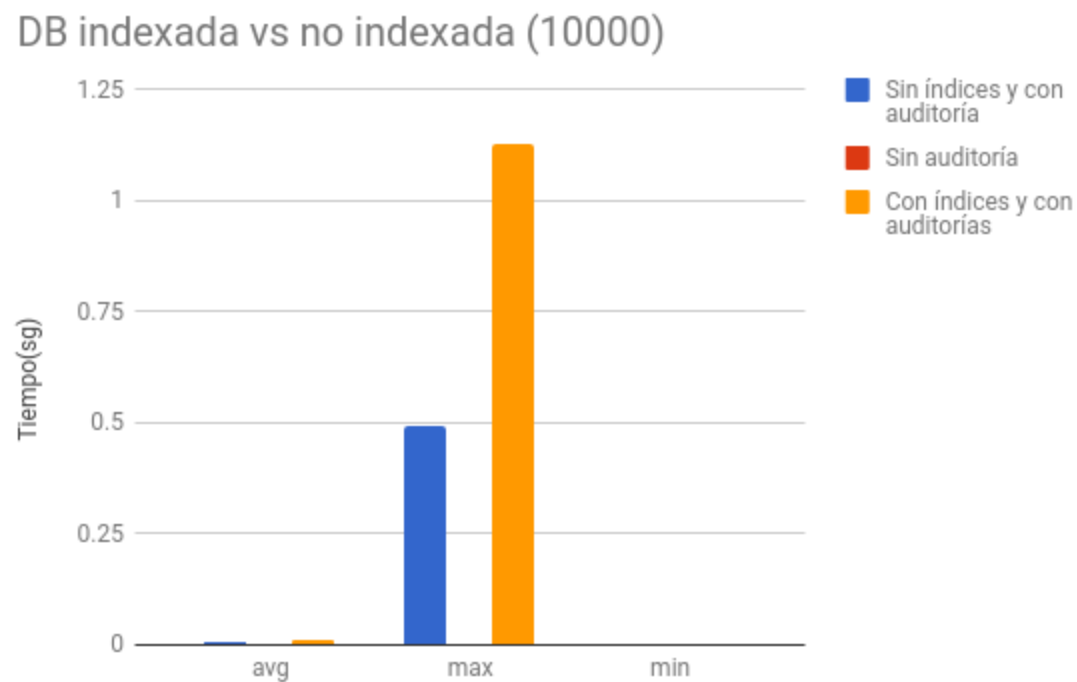


Figura 11. Resultados del promedio, mínimo y máximo para la tabla indexada comparada con la no-indexada al cargar diez mil registros.

El máximo con auditorías y sin índices es 0.4914271832 segundos y el mínimo es 0.0006959438 segundos. Dado que estos valores son lejanos entre sí, se muestra que MySQL no tiene un desempeño consistente durante la pruebas cuando se incluye la traza de auditoría. Ocurre de igual manera al agregar los índices.

DB indexada vs no indexada (100000)

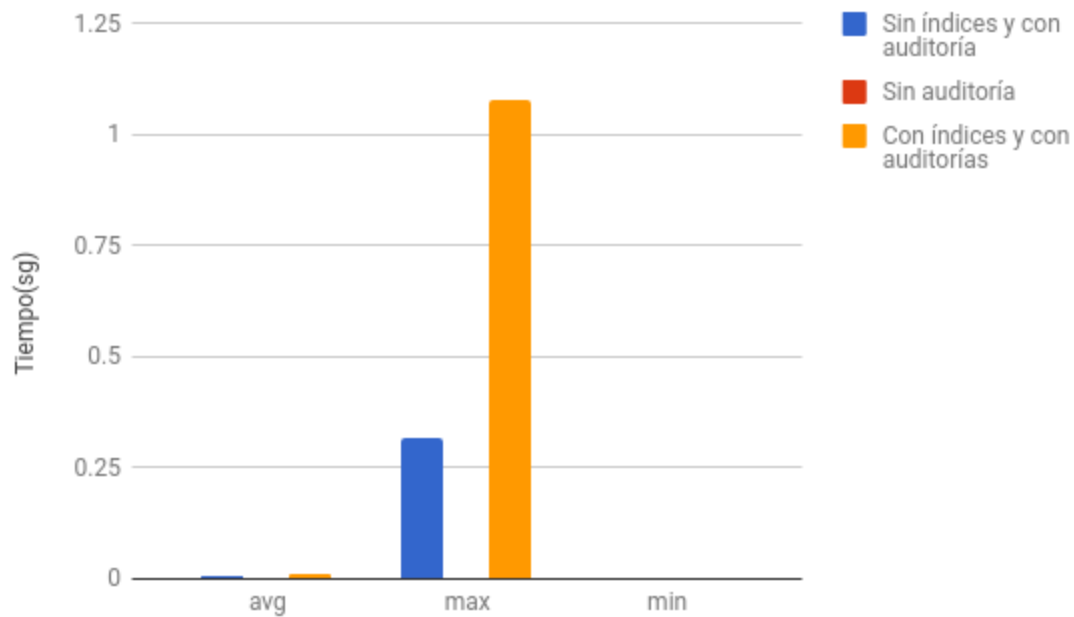


Figura 12. Resultados del promedio, mínimo y máximo para la tabla indexada comparada con la no-indexada al cargar cien mil registros.

El porcentaje de crecimiento de la tabla indexada sobre la no indexada, en este caso, es del 90%. El primero, incrementa el tiempo de inserción en 3161% y el segundo en 1612% sobre el registro sin auditorías.

DB indexada vs no indexada (1000000)

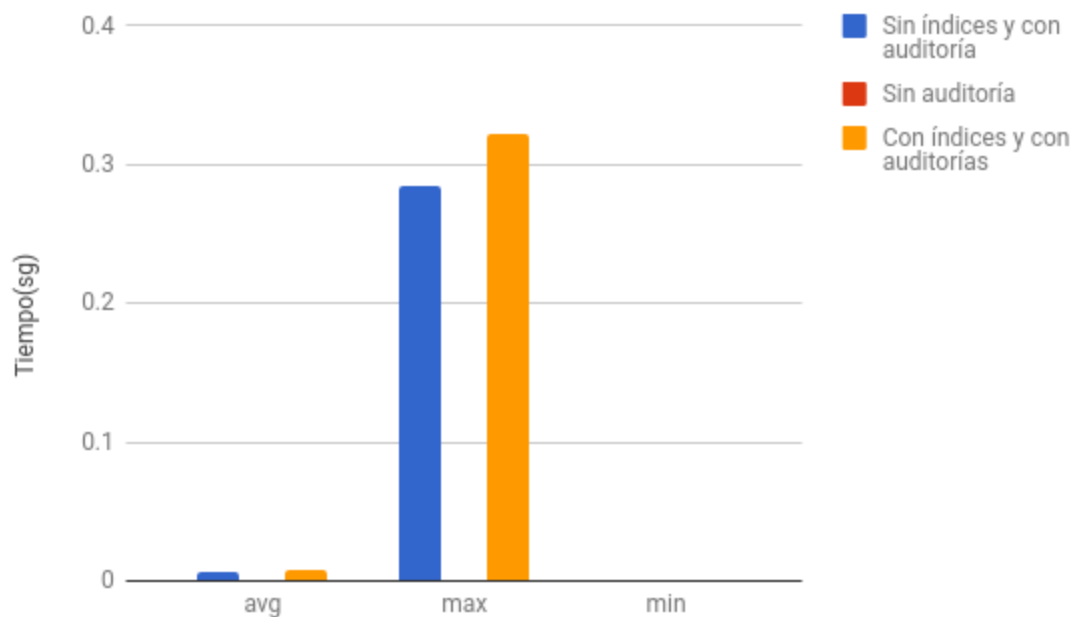


Figura 13. Resultados del promedio, mínimo y máximo para la tabla indexada comparada con la no-indexada al cargar un millón de registros.

Con este análisis se puede concluir que entre los manejadores de base de datos estudiados, MySQL muestra el peor desempeño al agregar trazas de auditorías. Considerando la robustez y confiabilidad de PostgreSQL, este es probablemente el mejor para llevar las auditorías con la librería Auditorías Turpial. Aunque SQLite presenta el menor tiempo de manera general, los índices prácticamente no afecta el tiempo.

Para caracterizar de manera más fiable el uso de índices, es necesario, en el futuro, realizar un estudio sobre las consultas.

Tabla 3. Resumen de los promedios de crecimiento usando MySQL.

Resumen		
%Crecimiento sin índices vs sin auditorías	%Crecimiento con índices vs sin auditorías	% Crecimiento sin índices vs con índices
1894.912972	2968.319394	53.66512553

Resultados obtenidos

SQLite

N	Sin índices y con auditoría	Sin auditorías	Con índices y con auditorías
Carga de la base de datos: 1000 registros			
sum	0.060614109	0.0194034576	0.0743262768
avg	0.0006061411	0.0001940346	0.0007432628
max	0.0008399487	0.0002629757	0.0020010471
min	0.0005831718	0.0001890659	0.000590086
Carga de la base de datos: 10000 registros			
sum	0.0608172417	0.0193741322	0.0748589039
avg	0.0006081724	0.0001937413	0.000748589
max	0.000838995	0.0002458096	0.0019161701
min	0.0005860329	0.0001890659	0.0005879402
Carga de la base de datos: 100000 registros			
sum	0.061671257	0.0196828842	0.0701274872
avg	0.0006167126	0.0001968288	0.0007012749
max	0.0008909702	0.0002541542	0.0022900105
min	0.0005850792	0.0001897812	0.0005891323
Carga de la base de datos: 1 millón de registros			
sum	0.0611720085	0.019516468	0.066955328
avg	0.0006117201	0.0001951647	0.0006695533
max	0.0010168552	0.0002639294	0.0017039776
min	0.0005850792	0.0001888275	0.0005939007

PostgreSQL

N	Sin índices y con auditoría	Sin auditorías	Con índices y con auditorías
Carga de la base de datos: 1000 registros			
sum	0.060614109	0.0194034576	0.1389353275
avg	0.0006061411	0.0001940346	0.0013893533
max	0.0008399487	0.0002629757	0.0043299198
min	0.0005831718	0.0001890659	0.0009920597

Carga de la base de datos: 10000 registros			
sum	0.0608172417	0.0193741322	0.1297955513
avg	0.0006081724	0.0001937413	0.0012979555
max	0.000838995	0.0002458096	0.00304389
min	0.0005860329	0.0001890659	0.0008819103
Carga de la base de datos: 100000 registros			
sum	0.061671257	0.0196828842	0.130531311
avg	0.0006167126	0.0001968288	0.0013053131
max	0.0008909702	0.0002541542	0.0027000904
min	0.0005850792	0.0001897812	0.0009260178
Carga de la base de datos: 1 millón de registros			
sum	0.0611720085	0.019516468	0.2921519279
avg	0.0006117201	0.0001951647	0.0029215193
max	0.0010168552	0.0002639294	0.1524419785
min	0.0005850792	0.0001888275	0.0009269714

MySQL

N	Sin índices y con auditoría	Sin auditorías	Con índices y con auditorías
Carga de la base de datos: 1000 registros			
sum	0.7648100853	0.0413048267	0.674380064
avg	0.0076481009	0.0004130483	0.0067438006
max	0.4487261772	0.0010368824	0.4603960514
min	0.0007300377	0.0002479553	0.0007231236
Carga de la base de datos: 10000 registros			
sum	0.5675678253	0.0261428356	1.258504391
avg	0.0056756783	0.0002614284	0.0125850439
max	0.4914271832	0.0007078648	1.126851082
min	0.0006959438	0.0002331734	0.0007381439
Carga de la base de datos: 100000 registros			
sum	0.6389715672	0.0373146534	1.217111826
avg	0.0063897157	0.0003731465	0.0121711183
max	0.3159770966	0.0013489723	1.078650951
min	0.000688076	0.0002319813	0.0007259846

Carga de la base de datos: 1 millón de registros			
sum	0.6576070786	0.0292971134	0.7514355183
avg	0.0065760708	0.0002929711	0.0075143552
max	0.2842690945	0.0007591248	0.3214919567
min	0.0007238388	0.000246048	0.0007269382