

Exposición:

Se ha desarrollado una base de datos de ejemplo llamada Testeo con el objetivo de mostrar parte de la funcionalidad de PostgreSQL, intentando darle un enfoque práctico lo más real posible.

La actividad se ha desarrollado y repartido lógicamente entre diversos scripts sql con el siguiente contenido:

- **Testeo_DDL_Functions:** contiene funciones, algunas de ellas parametrizables, para la generación y llenado de una base de datos sencilla para la gestión de la venta de productos.
- **Testeo_PrepereDB:** contiene llamadas a la funciones del script anterior para crear la base de datos con o sin índices, con o sin restricciones y con diferentes niveles de carga de datos para pruebas de escalabilidad.
- **AddingFunctionality:** contiene tablas para la gestión del stock, promociones y carritos de compra almacenados mostrando el uso de tablas unlogged. Aporta vistas materializadas y su gestión a través de desencadenadores. Se crean tablas temporales para los carritos de compra así como los desencadenadores y funciones necesarios para toda la implementación de las reglas de negocio básicas en la venta online de productos. Se introduce el uso del cursor y sus posibilidades de creación y uso.
- **Testeo_Auxiliary_Functions:** contiene funciones parametrizadas para la resolución de consultas con un alto impacto en el rendimiento y que ya han sido optimizadas.
- **Testeo_Demos:** contiene pruebas alternativas de diseño y uso de otros objetos, algunos se incorporan al diseño final y otros se descartan tras las pruebas de rendimiento.

Funcionalidad prevista para la base de datos

La conexión a la base de datos, una vez creada, será a través de un usuario que solo tenga permisos sobre funciones y vistas (materializadas o no), pero sobre ninguna tabla. La consulta a las promociones del día, por ejemplo, está optimizada gracias a una vista materializada.

Se han preparado funciones para ser llamadas por la aplicación que controle la venta, de manera que para cada usuario se crea una tabla temporal específica, con sus triggers preparados para la gestión del carrito de la compra de manera que las acciones del usuario sobre el carrito queden reflejadas en la reserva de unidades de producto en la tabla de stock. Una vez el usuario confirme o rechace el carrito se hará la gestión definitiva para la inserción de los registros en la tabla de detalles de venta y la actualización definitiva del stock.

Todo el proceso se realiza con los objetos propios de PostgreSQL.