



# Bases de Datos II

## Trabajo Práctico Integrador: Etapa 2

Fecha de entrega: 20/4

En este trabajo se implementarán consultas HQL para satisfacer los tests de la clase `DBliveryStatisticsTestCase`. Este *test case* inicializará la base de datos mediante la clase `DBInitializer` provista. Esta clase utilitaria creará un set de datos de pedidos, productos, usuarios, etc. para que las consultas obtengan resultados. Cabe destacar que, para crear los objetos de prueba, `DBInitializer` utilizará los métodos del servicio creado en el TP1, `DBliveryServiceImpl` (o el nombre que se haya elegido para implementar la interfaz `DBliveryService`).

La especificación de las consultas a implementar están descritas en la interfaz `DBliveryStatisticsService`, la cual pasará a ser interfaz padre de `DBliveryService`, implicando que `DBliveryServiceImpl` deberá agregar una implementación para cada uno de sus métodos. Estos métodos están descriptos usando el estándar Javadoc y, a su vez, los resultados esperados de las queries luego de haber inicializado la base de datos con `DBInitializer` estarán validados por `DBliveryStatisticsTestCase`. Para aprobar esta etapa, es requisito que todos los tests descritos en `DBliveryStatisticsTestCase` pasen.

Todo lo necesario para pasar a la siguiente etapa se encuentra en el repositorio base, ahora modificado:

<https://github.com/juliangrigera/bdlivery>

No hace falta volver a crear el proyecto, basta con agregar/reimportar los archivos alterados. Las diferencias con el repositorio base previo son:

- se modificó **pom.xml** para dependencias adicionales.
- se agrega la interfaz `DBliveryStatisticsService` al paquete **services**
- la interfaz `DBliveryService`
  - ahora extiende `DBliveryStatisticsService` y
  - especifica nuevos métodos para fijar la fecha de inicio en los **cambios de estado** y en la **creación de productos**.
- se agrega el paquete **utils** con la única clase `DBInitializer`
- se agrega la clase `DBliveryStatisticsTestCase`

La estructura del proyecto fue corregida respecto de la versión anterior para respetar la estructura de directorios y paquetes.



Los métodos a implementar de la interfaz **DBliveryStatisticsService** son los siguientes:

1. `getAllOrdersMadeByUser(String username);`  
Obtiene todas las órdenes realizadas por el usuario con username **username**
2. `getUsersSpendingMoreThan(Float amount);`  
Obtiene todos los usuarios que han gastando más de amount en órdenes en la plataforma
3. `getTopNSuppliersInSentOrders(int n);`  
Obtiene los **n** proveedores que más productos tienen en órdenes que están siendo enviadas
4. `getTop10MoreExpensiveProducts();`  
Obtiene los 10 productos más costosos
5. `getTop6UsersMoreOrders();`  
Obtiene los 6 usuarios que más cantidad de órdenes han realizado
6. `getCancelledOrdersInPeriod(Date startDate, Date endDate);`  
Obtiene todas las órdenes canceladas entre dos fechas
7. `getPendingOrders();`  
Obtiene el listado de las órdenes pendientes
8. `getSentOrders();`  
Obtiene el listado de las órdenes enviadas y no entregadas
9. `getDeliveredOrdersInPeriod(Date startDate, Date endDate);`  
Obtiene todas las órdenes entregadas entre dos fechas
10. `getDeliveredOrdersForUser(String username);`  
Obtiene todas las órdenes entregadas por el repartidor con username **username** en los últimos 10 días
11. `getSentMoreOneHour();`  
Obtiene las órdenes que fueron enviadas luego de una hora de realizadas
12. `getDeliveredOrdersSameDay();`  
Obtiene las órdenes que fueron entregadas el mismo día de realizadas
13. `get5LessDeliveryUsers();`  
Obtiene los 5 repartidores que menos órdenes entregaron
14. `getBestSellingProduct();`  
Obtiene el producto con más demanda



15. `getProductsOnePrice()` ;  
Obtiene los productos que no cambiaron su precio
16. `getProductIncreaseMoreThan100()` ;  
Obtiene la lista de productos que han aumentado más de un 100% desde su precio inicial
17. `getSupplierLessExpensiveProduct()` ;  
Obtiene el proveedor con el producto de menor valor histórico de la plataforma
18. `getSuppliersDoNotSellOn(Date day)` ;  
Obtiene los proveedores que no vendieron productos en un **day**
19. `getSoldProductsOn(Date day)` ;  
Obtiene los productos vendidos en un **day**
20. `getOrdersCompleteMorethanOneDay()` ;  
Obtiene las órdenes que fueron entregadas en más de un día desde que fueron iniciadas(status pending)
21. `getProductsWithPriceAt(Date day)` ;  
Obtiene el listado de productos con su precio a una fecha dada
22. `getProductsNotSold()` ;  
Obtiene la lista de productos que no se han vendido
23. `getOrderWithMoreQuantityOfProducts(Date day)`  
Obtiene la/s orden/es con mayor cantidad de productos ordenados de la fecha dada