

# INF-239 Bases de datos

## Tarea 1

Ayudante: Fernando Llorens  
fernando.llorens.13@sansano.usm.cl

17 de abril de 2019

### 1 Objetivos

- Conocer y aplicar las funcionalidades presentes en el lenguaje de base de datos SQL.
- Realizar métodos para poblar bases de datos de forma eficiente.

### 2 Descripción del problema

Suponga que usted trabaja para una conocida tienda de retail llamada *Sansanoplay*, la cual se dedica a la venta de videojuegos, accesorios para consolas y otros artículos *geek*. Durante los últimos años, el gerente general ha entablado buenas relaciones con *Nintendo*, por lo que ésta le facilita su base de datos con los videojuegos publicados hasta la fecha, con el fin de poder tener un mejor manejo de información al momento de que algún cliente necesite asesoría sobre su compra.

La base de datos actual de *Sansanoplay* consta de una tabla con los siguientes campos:

- Nombre del videojuego (String)
- ID del producto (int)
- Precio (int)
- Stock del videojuego (int)
- En bodega (int)
- Vendidos (int)

Mientras que la base de datos entregada por Nintendo consta de una tabla con los siguientes campos:

- Nombre del videojuego (String)
- ID del juego (int)
- Géneros (String)
- Desarrolladores (String)
- Publicadoras (String)
- Exclusividad (Booleano)
- Fecha de publicación (Date)
- Ventas globales (int)
- Rating (int)

El unificar ambas bases de datos <sup>1</sup> le permitirá tener un mejor manejo de información para los vendedores. Por ende, desde esta nueva plataforma, se podrán realizar ventas, actualizar stock, eliminar productos y ver detalles de cada uno.

Un detalle importante a tener en cuenta es que al realizar ventas, si un item posee menos de 10 unidades, se deberá chequear su stock en bodega y determinar si es posible reponer la cantidad justa para mantener esta condición.

### 3 Reglas de la entrega

1. Se deberá realizar una aplicación standalone (de escritorio, no es necesaria una interfaz de usuario) capaz de poder cargar los archivos csv y realizar *joins* entre ambas bases de datos, además de poder generar datos aleatorios para algunas columnas que se encontrarán en blanco.
2. También se deberá dar soporte a acciones del tipo CRUD (Create, Read, Update y Delete).
3. Finalmente, su aplicación deberá ser capaz de recibir consultas sobre: los 5 exclusivos más caros, los 3 géneros más vendidos (localmente y globalmente), las 3 desarrolladoras con más ventas locales y listar los juegos con mejor rating ordenados por fecha de lanzamiento.

*Hint* : Recuerde el uso de *Vistas* y *Triggers* para su implementación.

### 4 Restricciones y sugerencias

- **Toda interacción con la base de datos debe ser realizada con Querys dentro del código a implementar.**
- Su script para la carga de datos **puede** estar desacoplado de la aplicación. Luego, la aplicación se utiliza exclusivamente para la realización de consultas según los parámetros que se le ingrese.
- Se debe utilizar *Oracle DataBase Server 12C*.

---

<sup>1</sup>Se le facilitarán dos archivos csv para el desarrollo del problema

- Es **altamente recomendable** el uso de un sistema Linux en máquina virtual para el desarrollo, de igual forma pueden desarrollar en Windows.
- Se sugiere el uso de *Python*, *Java*, *.Net*, *C* o *C++*. Estas tecnologías son compatibles con *Oracle Database Server 12c*.
- *Visual Studio Code* y *Pycharm* son buenos IDE para desarrollar.
- Realizar la conexión con **odbc**, según el lenguaje que haya decidido utilizar.

## 5 Reglas del juego

La tarea debe cumplir con las siguientes reglas, de lo contrario podría sufrir un descuento final:

1. La tarea debe realizarse en equipos de **máximo 3 personas**. Los equipos deberán registrarse en moodle.
2. Los grupos son de carácter semestral: una vez conformados no podrán cambiar de compañeros. De existir algún problema, deben dirigirse directamente al ayudante de tareas.
3. La tarea debe entregarse en un archivo **.zip** vía moodle en la sección **Entregas** a más tardar el día **martes 14 de mayo a las 23:55** con el nombre **T1\_ROL1\_ROL2\_ROL3**.
4. La entrega deberá incluir un archivo **README** indicando los nombres de cada integrante del grupo y las instrucciones para la ejecución.
5. Habrá un descuento de 10 puntos de la primera hora de atraso, luego sumarán 20 puntos por día de atraso.
6. La defensa de la tarea se realizará en una fecha a convenir en moodle entre **el 15 y 17 de mayo**. Al momento de la defensa deberán estar todos los integrantes del equipo y tener todo lo necesario para la ejecución de esta.
7. Los integrantes deberán ir preparados para una **ronda de preguntas** sobre la tarea.