



## TALLER N°1 - PROGRAMACIÓN AVANZADA

**Docentes: Loreto Telgie Bendek – Cristian Chiang Ramírez – Carlos Luco Montofré.**

---

Fecha Entrega del Enunciado : Martes 22 de agosto.

Fecha de Término : Jueves 14 de septiembre. Hasta las 23:59 hrs.

---

### DOMINIO DEL PROBLEMA

La tienda de juegos *Zmart* necesita obtener información acerca de los juegos que ha vendido y de las categorías de juegos más solicitadas durante el mes de agosto. Estos registros ya fueron almacenados en un archivo de texto llamado **juegos.txt**, donde la tienda almacenó la siguiente información:

Estructura del archivo **juegos.txt**:

códigoJuego, nombreJuego, añoLanzamiento, precioVenta, porcGanancia y cantidadVendida.

Por ejemplo, el primer registro del archivo **juegos.txt** es el siguiente:

PS4G0658, Dark Souls III, 2016, 37000, 40, 26.

Código	Nombre	Año	P. Venta	% Ganancia	Cantidad vendida
--------	--------	-----	----------	------------	------------------

La ganancia por cada copia vendida de 'Dark Souls III' es de \$14.800 (porcentaje de ganancia respecto del precio de venta). Como se vendieron 26 unidades del juego, entonces Dark Souls III le proporciona a la tienda una ganancia de \$384.800.

La tienda *Zmart* desea saber que juegos son los que más vende, pero además desea saber cuáles son las categorías de juegos que más le generan ganancias, por ejemplo, le convendrá más comprar juegos de acción que de pelea (según la información recopilada del mes de agosto).

La tienda tiene definido que cada juego puede pertenecer a una o más categorías (como máximo se puede clasificar en 10 categorías). Esta información está registrada en el archivo **categorías.txt**.

Cada registro del archivo **categorías.txt** posee la siguiente estructura:

códigoJuego, nombreCategoría.

Por ejemplo, el juego Dark Souls III, de código PS4G0658, tiene asignado las categorías:

PS4G0658, Un jugador

...

PS4G0658, Acción

Se puede apreciar que no necesariamente los registros siguen un orden específico.

## REQUISITOS FUNCIONALES

La tienda Zmart requiere que los consultores informáticos generen un pequeño sistema en Java (por consola, nada de interfaz gráfica) que cumpla con los siguientes requisitos funcionales.

**RF1:** Dado un año de lanzamiento ingresado por consola, imprimir por esta misma: a) el nombre del juego y la cantidad de unidades vendidas, para el juego que más se ha vendido. b) El nombre del juego y la ganancia obtenida, para el juego que ha generado mayor ganancia.

**RF2:** Dada una o más categorías ingresadas por consola, imprimir por consola la ganancia total por ventas generada por todos los juegos que pertenezcan al menos a una de las categorías ingresadas.

**NOTA:** Como un juego puede pertenecer a más de una categoría, tenga cuidado de no repetir más de una vez la ganancia que ha generado ese juego.

**RF3:** Dado el nombre de una categoría, se debe generar el archivo *top-ten-<categoría-ingresada>.txt* que contenga el *top-ten* de los juegos que más se han vendido. Por cada juego de los 10 mejores debe registrarse el nombre y la cantidad vendida. Estos datos deben ser listados ordenados de mayor a menor según la cantidad vendida.

**Ejemplo:** Si ingresa “Lucha” se genera el archivo *top-ten-Lucha.txt* con los 10 primeros juegos que más se han vendido, ordenados de mayor a menor.

**RF4:** Generar un menú con opciones para cumplir con RF1 - RF3 y una opción para finalizar la ejecución del programa.

## CONSIDERACIONES

- Se les proporcionará los archivos de texto con los que trabajará
- Ambos archivos de texto entregados tendrán sus datos separados por una coma (,).
- **Los archivos de texto solo se deben leer una vez.**
- El taller debe ser realizado en grupos de 2 alumnos del mismo paralelo.
- Para la entrega 1 y 2 se debe publicar en la plataforma el informe en formato .pdf y entregar impreso.
- Para la entrega 3 se debe publicar en la plataforma el proyecto (exportar como .zip), el cual llevará por nombre el o los apellidos del o los integrantes. Por ejemplo: castro.rojas.zip
- El nombre del Proyecto se denominará Taller1.
- Se debe hacer uso de subprogramas (funciones y/o procedimientos).
- La entrega 3 debe realizarse en NetBeans o Eclipse.
- El taller deberá estar documentado. Se recomienda utilizar el estándar Java Doc.
- Las consultas se realizarán al correo del ayudante. (**Ayudante:** José Espinoza, **correo:** jes004@alumnos.ucn.cl)
- Cada incumplimiento será sancionado con 0,3 de la calificación final.
- En caso de copia (códigos completamente iguales) la nota será 1.0 para todos los que entregaron el mismo código y el caso se presentará ante el consejo.

## FECHAS DE ENTREGA

▪ **Entrega N° 1: 28 de agosto (23:59)**, Diseño de la estructura para los datos.

Se debe entregar un informe (pdf e impreso) con las estructuras de datos que van a utilizar, es decir, identificación de arreglos, vectores o matrices van a ocupar, que tipo de información van a almacenar, cómo se relacionan unos arreglos con los otros. Diagrame un ejemplo con datos de prueba que servirán para resolver los RF planteados.

**EL INFORME DEBE ENTREGARSE IMPRESO EN SECRETARÍA TERCER PISO DE LA ESCUELA DE INGENIERÍA.**

La revisión va a considerar

- Cumplimiento del formato descrito en el archivo *PautaCorrecciónTalleres.docx*
- Por cada falta de ortografía se van a descontar 0,2 décimas (Si tuvieron un 7.0 y una falta, entonces tendrán un 6.8)

▪ **Entrega N° 2: 01 de septiembre (23:59)**. Diseño de la estructura del programa.

Se debe entregar un informe (pdf e impreso) de la estructura del programa por la cual se va a regir la codificación, en otras palabras, la identificación del programa principal y las rutinas del sistema. Establecer cuál será el flujo de ejecución (por ejemplo, primero se ejecuta esto, luego esto), cómo se relaciona cada sección de la estructura del programa con la estructura de datos de la Entrega N°1. **NO SE PIDE SEUDOCÓDIGO NI CÓDIGO.** *(se deben agregar diagramas, esquemas, comentarios entre otros, que representen de mejor manera la estructura del programa).*

**EL INFORME DEBE ENTREGARSE IMPRESO EN SECRETARÍA TERCER PISO DE LA ESCUELA DE INGENIERÍA.**

La revisión va a considerar

- Cumplimiento del formato descrito en el archivo *PautaCorrecciónTalleres.docx*
- Por cada falta de ortografía se van a descontar 0,2 décimas.

▪ **Entrega N° 3: 14 de septiembre (23:59)**. Código de la aplicación.

Se debe subir a plataforma un archivo .zip que contemple la **carpeta del proyecto** donde está almacenado todo el código fuente de la aplicación realizado en NetBeans o Eclipse.

La revisión va a considerar

- Si el archivo subido no es un .zip o un .rar (o similar), la nota máxima es un 2.0
- El .zip o .rar debe contener la **carpeta del proyecto** (comunicarse si tiene dudas), en el caso que únicamente el archivo .java, entonces la nota máxima será un 4.0
- Si el código no compila, la nota máxima es un 2.0