



# UNITEC

## Facultad de Ingeniería

### Jefatura de Ingeniería en Sistemas

### Computacionales

#### Laboratorio de Programación III

#### Proyecto

Instructor: José Aceituno



**unitec**<sup>®</sup>  
LAUREATE INTERNATIONAL UNIVERSITIES<sup>®</sup>



Enhorabuena emprendedor, usted acaba de iniciar una empresa de desarrollo de software y su primer trabajo es realizar un sistema de inventario para un amigo que puso una tienda de videojuegos y consolas usadas, a partir de este momento vamos a tratar a su amigo como el PO (Product Owner).

Para empezar, debemos saber que vende la tienda. La tienda vende **consolas** solo de las compañías de **Microsoft**, **Sony** y **Nintendo**.

El PO desea saber de cada consola el *año* en que salió, el *modelo*, el *estado* en que se encuentra, el *número de serie* y el *precio*. Cabe mencionar que como en cualquier base de datos, el número de serie debe ser único, por lo que deberá haber una validación ahí.

**Nota:** La consola no lleva ningún campo que diga “cantidad”, pero si deseo saber cuantas consolas de cada modelo tengo, cada consola tiene una serie diferente por lo que el objeto consola es independiente de todas las demás consolas, debe haber una forma de saber cuantas hay en inventario. También, debo poder agregar n cantidad de consolas del mismo modelo a la vez, pero por cada una se me debe preguntar el número de serie.

Los modelos de consola que hay por clase son:

**Microsoft:**

- xbox
- xbox 360
- xbox One

**Sony:**

- Play Station 1
- Play Station 2
- Play Station 3
- Play Station 4
- PSP
- PS Vita

## Nintendo:

- Nintendo Entertainment System
- Super Nintendo Entertainment System
- Nintendo 64
- Nintendo Gamecube
- Nintendo Wii
- Nintendo Wii U
- Nintendo Switch
- Gameboy
- Gameboy Colors
- Gameboy Advance
- Nintendo DS
- Nintendo DSi
- Nintendo 3DS
- Nintendo New 3DS

Con eso se concluye la parte de las consolas.

En cuanto a los **videojuegos**, por razones legales no se pueden vender videojuegos de todas las compañías, solo de las siguientes: **Microsoft, Sony, Nintendo, Bandai, Konami, Square Enix., Electronic Arts, SEGA, Ubisoft.**

El PO desea saber de cada juego el *nombre*, el *año* en que salió, para que *consola*, el *número de jugadores* que lo pueden jugar, el *género*, el *estado* en que se encuentra, el *número de serie* y el *precio*. Cabe mencionar que como en cualquier base de datos, el número de serie debe ser único, por lo que deberá haber una validación ahí.

Existen 2 tipos de **usuario**:

El **usuario administrador** que puede agregar, modificar y eliminar cualquier cosa (NO PUEDE VENDER). De este se conoce *nombre de usuario y contraseña*.

El **usuario vendedor** que solo puede agregar a inventario y vender. De este solo se conoce el *nombre*, no tiene contraseña ya que son usuarios temporales, *hora de entrada al sistema*, *hora de salida del sistema*.

Al entrar en la función de vender, el usuario deberá elegir si quiere vender consola o videojuego, luego dependiendo debe elegir de que compañía quiere la consola o el videojuego y así sucesivamente. Cada **venta** tiene el *nombre del cliente*, una *lista de consolas* y *lista de juegos*, aparte de una *hora de finalización*, el nombre del *usuario* que la realizó y un *subtotal*.

Al momento que el usuario ya no tenga deseos de comprar nada más, se debe cerrar la venta y al hacer cierre, en el directorio del programa debe haber un subdirectororio que se llame *log\_ventas*, en el cual se debe crear un archivo por cada venta y el nombre del archivo deberá ser la fecha y hora del cierre de la venta (con extensión .log). El contenido de este archivo debe ser algo así:

```

                                GAMEHUB
FECHA:      26/02/2017
HORA:       22:07:46
VENDEDOR:   USUARIO
CLIENTE:    NOM_CLIENTE

CANTIDAD ARTÍCULOS N

NOMBRE ARTÍCULO 1    PRECIO
NOMBRE ARTÍCULO 2    PRECIO
.
.
.
.
NOMBRE ARTÍCULO N    PRECIO

SUBTOTAL:
IMPUESTO: (15%)
TOTAL:
```

Por último, al momento de que un usuario vendedor cierre el programa, en el directorio principal del programa debe haber una carpeta llamado *usuarios\_log* en la cual se van a guardar vitácoras de usuario, de nombre van a llevar este formato *NombreVendedor\_fecha.log* y el contenido:

```
GAMEHUB

NOMBRE:  VENDEDOR
HORA ENTRADA:  HORA_E
HORA SALIDA:   HORA_S

CANTIDAD ARTÍCULOS VENDIDOS: #
DINERO GENERADO:   #
```

Debe incluir ya por defecto una base de datos de almenos 40 consolas y 120 juegos.

## Ponderación

Elemento	Valor
Clases y makefile	2
Funciones de usuario administrador (agregar, modificar y eliminar)	3
Función de usuario vendedor	3
Uso de GitHub	1
Validaciones de números de serie	1
Facturación y log de empleado	5
Leer base de datos de archivo binario	x0.5

## Penalizaciones

Habrán penalizaciones por:

- Uso de variables globales
- Variables no significativas
- No usar Git/GitHub
- Mala tabulación
- Segmentation fault, floating point exception, what() bad alloc, entre otros errores que quiebren el programa.
- Interfaz no amigable.

## Posibles preguntas

### ¿qué significa la última fila de la ponderación?

Significa que si no leen los datos d archivo binario y cada vez que inicie el programa tengo que agregar videojuegos y consolas, multiplico la nota total del proyecto por 0.5.

### No hemos visto bases de datos, ¿cómo incluimos una en el proyecto?

Por base de datos me refiero a los archivos de texto de los cuales van a leer los datos.

### No se usar archivos.

Esa es la parte de investigación del proyecto.

### Son demasiadas clases, Nintendo tiene 14 clases hijas.

No, por facilidad de ustedes, todas las qque son clases están en negrita y todo lo que está en cursivo son atributos, por lo tanto la lista de las consolas de nintendo no son clases hijas, si no solo una lista de posibles elementos.

### ¿Podemos meter otras consolas que no están incluidas como las de SEGA?

Si quieren, yo solicito que todo lo que está en el documento esté en el programa, si ustedes quieren agregar clases, funciones, elementos, integraciones, librerías o cualquier cosa, son libres de hacerlo.