**ISAE UNIVERSIDAD**

**TECNICO EN INFORMATICA**

**PROGRAMACION VI**

**Tema:**

**Informe sobre Software lenguaje en Java**

**Profesor:**

**Ibrain Lin**

**Estudiante:**

**Madelin Pérez 8-891-926**

**Grupo:**

**METETI04**

**Año Académico:**

**2016**

# INTRODUCCION

**Java** es un lenguaje de programación de propósito general, concurrente, orientado a objetos que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible.

El lenguaje de programación Java fue originalmente desarrollado por James Gosling de Sun Microsystems (la cual fue adquirida por la compañía Oracle) y publicado en 1995 como un componente fundamental de la plataforma Java de Sun Microsystems. Susintaxis deriva en gran medida de C y C++, pero tiene menos utilidades de bajo nivel que cualquiera de ellos. Las aplicaciones de Java son generalmente compiladas a bytecode (clase Java) que puede ejecutarse en cualquier máquina virtual Java (JVM) sin importar la arquitectura de la computadora subyacente.

**Elabore un informe sobre un software en lenguaje Java. el informe debe tener.**

## 1. Código fuente

```
2. package ejerc03;
3. import leer.*;
4. public class Test {
5. 
6.   public static void main(String[] args) {
7.     /*
8.      * Sólo se debe añadir métodos para establecer los valor
   es de los atributos, poder imprimirlos datos en pantalla y ca
   lcular cantidad a pagar según el precio de venta. De momen
   to, no hace falta gestionar el almacén con los artículos que
   quedan, sólo se pedirá la cantidad de artículos que lleva el
   cliente y se dará el precio a pagar según esa cantidad.
9.      */
10. 
11.         System.out.println("Bienvenido a la tienda de JUEG
   OS, MÚSICA Y PELÍCULAS\n--------\n"+
12.                 "El programa simula una tienda que vende ju
   egos, música y películas\n"+
13.                 "Los artículos sólo se pueden vender si está
   n disponibles en la tienda\n"+
14.                 "El usuario puede:\n"+
15.                 "\t Comprar productos de los existentes.\n"+
16.                 "\t Añadir nuevos productos a la cesta.\n"+
17.                 "\t Consultar el importe de la cuenta.");
18. 
19.         boolean continuar = true;
20.         int lecturaProducto, lecturaCantidad; //Variables par
   a seleccionar el producto y la cantidad que se quiere compr
   ar
21. 
22.         //Se instancian y cargan los productos
23.         Producto disco1 = new Musica("Portishead", 18.53,
   6, true, "Trip Hop");
24.         Producto disco2 = new Musica("Radiohead", 21.2, 2
   0, true, "Rock");
25.         Producto cine1  = new Cine("Hierro 3", 30.2, 5, true
   , "Kim Ki Duk");
26.         Producto juego1 = new Videojuego("Mario Bross", 3
   5.11, 0, false, "Plataformas");
27. 
28.         //Se crea el array "catálogo" para contener los prod
   uctos. Su dimensión viene del número de veces que se
29.         //instancia el contructor de Producto
```

```
30.         Producto catalogo[] = new Producto[Producto.dimes
   ionArray];
31.         //Se crea el objeto gestion para trabajar (mostrar y
   vender productos, y mostrar caja)
32.         Gestion gestion = new Gestion();
33.
34.         //Se rellena el array catálogo
35.         catalogo[0] = disco1;
36.         catalogo[1] = disco2;
37.         catalogo[2] = cine1;
38.         catalogo[3] = juego1;
39.
40.         do {
41.           System.out.println("\n\nIntroduzca la opción que d
   esea realizar:\n"
42.                   +"1. Mostrar productos\n"
43.                   +"2. Vender productos\n"
44.                   +"3. Mostrar caja\n"
45.                   +"SALIR --> Pulse cualquier otro número\n"
46.                   );
47.           switch (Leer.datoInt()) {
48.             case 1:
49.               gestion.mostrarProductos(catalogo);
50.               break;
51.             case 2:
52.               System.out.println ("¿Que producto desea comp
   rar?");
53.               gestion.mostrarNombreProductos(catalogo);
54.               lecturaProducto=Leer.datoInt();
55.               System.out.println("¿Cuánta cantidad desea co
   mprar?");
56.               lecturaCantidad=Leer.datoInt();
57.               //Se carga el producto y la cantidad solicitada p
   or el usuario
58.               gestion.comprarProducto(catalogo, lecturaProd
   ucto, lecturaCantidad);
59.               break;
60.             case 3:
61.               System.out.println(gestion.mostrarCaja() +" €");
62.               break;
63.             default:
64.               //Se sale del programa
65.               continuar=false;
66.           }
67.
68.         } while(continuar);
69.
70.         System.out.println ("---- Gracias por usar la aplicaci
   ón. ----");
71.
```

```
72.         }
73.
74.     }
```

Explicando un poco el código de Test. Se declaran tres variables. Una variable booleana "*Continuar*" para repetir el proceso de compra, dos variables enteras "*lecturaProducto*" y "*lecturaCantidad*" para que el usuario elija el producto que desea y su cantidad. Se declaran y cargan cinco arrays para contener productos de tipos "*Música*", "*Cine*" y "*Juegos*" que serán guardados en el array "*Catálogo*". Se declara el objeto "*Gestion*" para realizar los procesos del programa.

Al ejecutarse el programa el usuario ve un pequeño menú con las opciones: Mostrar productos, vender, mostrar caja y salir. La opción elegida por el usuario determinará que trozo de código debe ejecutarse mediante un Switch.

Si la opción elegida es mostrar productos se le aplica el método**mostrarProductos (Catalogo)** al objeto "*Gestion*".

Si la opción es vender producto se muestra los nombres de los productos mediante **gestion.mostrarNombreProductos (catalogo)** y luego se solicita la cantidad de dicho producto, y se ejecuta la venta con el método: **gestion.comprarProducto (catalogo, lecturaProducto, lecturaCantidad)** en el que se pasan como parámetros el catálogo, el producto y la cantidad.

Si la opción es "mostrar caja" se ejecuta el métodogestion.mostrarCaja () que devuelve el importe de las ventas.

La última opción de salida se lleva a cabo si el usuario introduce un número diferente al 1, 2 ó 3.

La clase que lleva la lógica del programa es la Gestion, su código es el siguiente:

```
Package ejerc03;


public class Gestion {


  //Attributes

  Private Producto productos [] = null;

  prívate doublé caja;


  //Constructors

  public Gestion() { }
```

```
  public Gestion(Producto[] productos) {
    this.productos = productos;
  }


  //Methods
  public Producto[] cargarProductos() {

    return productos;
  }

    Public void mostrarProductos(Producto[] productos) {
        for (int i = 0; i < productos.length; i++) {
          System.out.print(productos[i]+"\n-------\n");
        }
    }
    public void mostrarNombreProductos(Producto[] productos) {
        for (int i = 0; i < productos.length; i++) {
          System.out.println (i+1 +" "+productos[i].getNombre()+"\n");
        }
        System.out.println ("\n----------\n");
    }
  public double comprarProducto(Producto[] productos, int num, int
cantidadUnidades) {
    If (productos [num-1].isDisponible ()) {
      If (productos [num-1].getCantStock () >= cantidadUnidades){
        System.out.println("La compra se ha realizado con éxito!!\n");
        Productos [num -1].setCantStock (productos [num -1].getCantStock ()-
cantidadUnidades);
        Return caja+=cantidadUnidades*productos [num-1].getPrecioUnit ();
      } Else {System.out.println("No hay cantidad suficiente de producto");}
```

```
    } Else {System.out.println ("No hay cantidad suficiente de producto");}
    return caja;
  }
  public double mostrarCaja() {
    System.out.print("El total de la caja es ");
    caja=Math.round(caja*100);
    return caja/100;
  }
}
```

En la clase "**Gestion**" se declaran dos atributos, un array de tipo "*Producto*" para contener los productos y otro de tipo "*Double*" para contener el importe. Se declaran dos constructores, uno vació y otro que carga el array de productos.

Esta clase contiene 5 métodos.

El primero, **public Producto [] cargar Productos ()**, devuelve el array de productos.

El segundo, **public void mostrarProductos (Producto[] productos)**, recibe como parámetro el array de productos que se mostrarán.

El tercer método, **public void mostrarNombreProductos (Producto [] productos)**, es parecido al anterior, con la diferencia que dentro de su código se llama al código método getNombre () para imprimir por pantalla el nombre en lugar de todo el array de productos.

El cuarto método, public double **comprar Producto (Producto [] productos, int num, int cantidad Unidades)**, recibe el array de productos, un entero identificador del producto seleccionado y otro entero para la cantidad de productos elegida por el usuario. Este método, comprueba que el producto seleccionado exista **if (productos[num-1].isDisponible())** y que la cantidad seleccionada sea menor que la cantidad en stock **if (productos[num-1].getCantStock() >= cantidadUnidades)**. En el caso que la compra se pueda llevar a cabo cumpliendo las condiciones anteriores, la cantidad de productos en stock será actualizado **productos[num -1].setCantStock(productos[num -1].getCantStock()-cantidadUnidades)** y se añadirá el importe de

la compra a la caja: **caja+=cantidadUnidades*productos[num-1].getPrecioUnit()**.

El último método, **public double mostrarCaja()**, devuelve entero que indica el importe de la venta actual **caja=Math.round(caja*100); return caja/100;**.

En la clase Producto se definen atributos comunes a los tipos "*Música*", "*Videojuego*" y "*Cine*": "*nombre*", "*precioUnit*", "*cantStock*" y "*disponible*". Se crea un cuarto atributo estático llamado "*dimensionArray*" para que cuando en la clase Test se cree el catálogo de productos le podamos designar la dimensión de este array directamente usando esta variable: **Producto catalogo[] = new producto[Producto.dimesionArray]** (cada nueva creación de objeto Producto incrementa en uno la dimensión).

El código de la clase **Producto** es el siguiente:

Java

```
1  package ejerc03;
2
3  public abstract class Producto {
4
5    //Attributes
6    private String nombre;
7    private double precioUnit;
8    private int cantStock;
9    private boolean disponible = false;
10   public static int dimesionArray;
11
12
13   //Constructors
14   public Producto() { }
15   public Producto(String nombre, double precioUnit, int cantStock, boolean d
16 isponible) {
17     this.nombre   = nombre;
18     this.precioUnit = precioUnit;
19     this.cantStock  = cantStock;
20     this.disponible = disponible;
21
22     dimesionArray++;//Se obtiene con esta variable la dimensión del array. S
23 egún número de instancias del constructor
24   }
25
26
27   //Methods
28   @Override
29   public String toString() {
30     return  "Nombre: "      +this.getNombre()+    "\n"+
```

```
31        "Precio unidad: " +this.getPrecioUnit()+  " €\n"+
32        "En Stock: "    +this.getCantStock()+ "\n";
33  }
34
35  //Get and Set
36  public String getNombre() {
37    return this.nombre;
38  }
39  public void setNombre(String nombre) {
40    this.nombre = nombre;
41  }
42  public double getPrecioUnit() {
43    return this.precioUnit;
44  }
45  public void setPrecioUnit(double precioUnit) {
46    this.precioUnit = precioUnit;
47  }
48  public int getCantStock() {
49    return this.cantStock;
50  }
51  public void setCantStock(int cantStock) {
52    this.cantStock = cantStock;
53  }
54  public boolean isDisponible() {
55    if(getCantStock()>0) this.disponible = true;
56    return this.disponible;
57  }
58  public void setDisponible(boolean disponible) {
59    this.disponible = disponible;
60  }

    }
```

En esta clase generamos los *getters* y *setters* para acceder a los atributos. También el método **toString()**. Uno de los métodos generados**public boolean isDisponible()** lo modificamos un poco. En la declaración de atributos pusimos a "*false*" la disponibilidad, con lo que siempre figura que no existe producto, para ello usamos: **if(getCantStock()>0) this.disponible = true;** sabremos si hay disponibilidad.

Las tres clases restantes: "*Musica*", "*Videojuego*" y "*Cine*" se crean por *herencia* de la clase padre "*Producto*". Por darle un poco de diferencia, he añadido un atributo a cada una de ellas: "genero", "tematica" y "director". Por herencia se crean los constructores y método toString(). Y se añaden getters y setters de los nuevos atributos. Para no añadir mucho código, copio el código de la clase**Musica**:

Java

```
package ejerc03;

public class Cine extends Producto {

  //Attribute
  private String director;


  //Constructor
  public Cine() { }
  public Cine(String nombre, double precioUnit, int cantStock, boolean dispo
nible, String director) {
    super(nombre, precioUnit, cantStock, disponible);
    this.director = director;
  }


  //Methods
  @Override
  public String toString() {
    return  "ARTÍCULO DE CINE\n"+
        super.toString()+
        "Director: "  +this.director;
  }

  //Gets and Sets
  public String getDirector() {
    return director;
  }
  public void setDirector(String director) {
    this.director = director;
  }
}
```

**2. Objetivo del software**

Ejemplo de programa Java para tienda de venta de juegos, música y películas. El programa está compuesto de seis clases: Test.java, Gestion.java, Producto.java, Cine.java, Musica.java y Videojuego.java. La clase producto define propiedades comunes de los tipos Cine, Música y Videojuego, y cada una de estas define sus propios atributos. La clase Gestión, como su nombre indica, gestiona la tienda y la clase Test ejecuta el programa.

### 3. Icono del software de escritorio


### 4. Formulario del software


### 5. Cotización del software.

# Servicios Arlyn

Santa fe, Darién

299-7845

Nombre del cliente

**Gitanya Sánchez**

Nombre de la Empresa

**Servicios Madelin**

P r e s e n t e:

Software para tienda de Video Juegos, Cine, Película.

La presente propuesta abarca el diseño, desarrollo y mantenimiento del Software por dos años.

**Alcance del proyecto**

Se desarrollará un Software que calcula y guarda información del cliente acerca de las compras que va a realizar, con el objeto de proyectar una excelente imagen de la empresa del cliente al mundo a través del Sistema rápido.

**Concepto:**

**Precio:**

Diseño y Desarrollo de Software y mantenimiento con base de datos por dos años.

$ 5,500.00

**Total**

$ 5,500.00

Estoy a tus órdenes para cualquier duda o comentario.

Atentamente,

Madelin Pérez
Director General