

## PROGRAMACIÓN ORIENTADA A OBJETOS

### (Laboratorio de Prácticas)

Titulaciones de Grado en Ingeniería Informática, Ingeniería en Sistemas de Información e Ingeniería de Computadores.

## EJERCICIO LABORATORIO

---

Una empresa nos ha encargado el diseño y realización de una aplicación informática llamada “JavaPop” que permita la venta de productos entre particulares.

Los elementos a tener en cuenta en la aplicación son los siguientes:

#### **Usuarios de la aplicación.**

La aplicación puede ser utilizada por dos tipos de usuarios:

1. Administrador de la aplicación: Es el responsable de la aplicación, utilizará los siguientes datos para acceder: correo “*admin@javapop.com*” y clave “*admin*”. Podrá gestionar tanto a los usuarios como sus productos.
2. Cliente: Para poder acceder a la aplicación, tanto los compradores como los vendedores deberán estar registrados como usuarios. En el momento de acceder, la aplicación les permitirá registrarse o entrar, en caso de que sean usuarios registrados entrarán con su correo y su clave. En el momento de registrarse, el nuevo usuario deberá introducir al sistema los siguientes datos:
  - a. DNI.
  - b. Nombre.
  - c. Correo electrónico del usuario.
  - d. Clave.
  - e. Ubicación (código postal y ciudad).
  - f. Tarjeta de crédito (numero de 16 dígitos).
3. Cliente profesional: Representa un tipo especial de cliente, que además de los datos del cliente tendrá una serie de datos adicionales. Los clientes profesionales, generalmente poseedores de tiendas, se les cobra una cuota mensual **de 30€** que se les cargará en su tarjeta de crédito. A estos clientes se les permite la inserción de los siguientes datos:
  - a. Descripción.

- b. Horario de apertura.
- c. Teléfono.
- d. Web

Para el cobro de los 30€ solo es necesario avisar al usuario con un mensaje donde se le indica que se le ha hecho un cargo en su tarjeta de crédito (indicando su número).

### Productos en Venta.

Se deben almacenar los datos de cada uno de los productos disponibles para su venta. Los datos a considerar para cada producto son los siguientes:

- Título (resumen de lo que se vende con palabras clave).
- Descripción (información relevante del producto).
- Categoría:
  - Moda y accesorios
  - TV, audio y foto
  - Móviles y telefonía
  - Informática y electrónica
  - Consolas y videojuegos
  - Deporte y ocio
- Estado del producto
  - Nuevo
  - Como nuevo
  - Bueno
  - Aceptable
  - Regular
- Precio de venta del producto.
- Fotografía.
- Fecha de publicación.
- Ubicación del usuario que publica el producto.

Un usuario puede marcar un producto como “urgente” pagando 5€ y de esta forma su producto destacará sobre los demás durante 7 días.

Para el cobro de los 5€ solo es necesario avisar al usuario con un mensaje donde se le indica que se le ha hecho un cargo en su tarjeta de crédito (indicando su número).

### Búsqueda de productos

Para poder comprar un producto los usuarios podrán utilizar una herramienta de búsqueda. El usuario seleccionará una categoría y a continuación las palabras clave del producto que está buscando. Si no desea introducir ninguna palabra clave se le mostrarán todos los productos de esa categoría. El resultado de la búsqueda será una lista de los productos ordenados por proximidad a la ubicación del usuario. Tendrán prioridad los marcados como urgentes.

Para realizar la búsqueda se tendrá en cuenta las palabras clave, si un producto tiene al menos una de las palabras clave se le presentará al usuario.

Para calcular la distancia entre códigos postales se realizará de la siguiente forma:

Si coinciden los tres primeros dígitos se considera muy próximo, si coinciden los dos primeros se considera próximo.

### Compra del producto

En el caso de que el usuario seleccione un producto para su compra, el dueño del producto deberá marcarlo como vendido para completar el proceso. **Para realizarlo se simulará mediante un mensaje donde se pregunta al vendedor si está de acuerdo con la venta, si selecciona que SI se aprobará esa venta.**

Una vez confirmada la compra, la aplicación **guardará la venta** realizando automáticamente las siguientes operaciones:

- Generar un documento de venta en un fichero de texto. Deberá incluir los datos del producto, la fecha de venta, el DNI y nombre del vendedor y del comprador.
- Almacenar los datos de la venta: Fecha de la venta, datos del producto y el DNI y nombre de los usuarios implicados.
- Dar de baja el producto para que no aparezca en futuras búsquedas.

### Funcionamiento de la aplicación

Se debe realizar una aplicación gráfica con Swing que permita, una vez seleccionado el acceso como administrador o cliente, realizar las siguientes operaciones:

1. En caso de acceder como administrador las opciones disponibles serán:
  - 1.1. *Consulta de usuarios*: Se mostrará un listado de los usuarios de la aplicación pudiendo consultar su información. Si se detecta el uso fraudulento de un usuario en la aplicación podrá darlo de baja con todos sus productos.
  - 1.2. *Consulta de productos*: Se mostrará un listado de los productos pudiendo consultar su información. Si se sospecha que algún producto es fraudulento se podrá dar de baja.
  - 1.3. *Consulta de ventas realizadas*: Se irán mostrando las ventas, pudiendo seleccionar todas o solo a partir de una fecha dada.
2. En caso de acceder como cliente las opciones disponibles serán:
  - 2.1 Registro del cliente.
  - 2.2 Pago de cliente profesional con acceso a editar sus datos adicionales.
  - 2.3 Alta de nuevos productos.
  - 2.4 Baja de los productos que tiene registrados un usuario.
  - 2.5 Búsqueda de productos.
  - 2.6 Compra de productos.

**Cada vez que se arranque la aplicación, tendrá disponibles todos los datos almacenados desde la última vez que se utilizó. Del mismo modo, al terminar la sesión, deberán guardarse todos los datos actualizados. Se utilizarán para ello los archivos que sean precisos.**

## Requisitos

La aplicación estará escrita en lenguaje Java, compatible con Java SE. No deberán utilizarse clases o métodos obsoletos ("deprecated").

La aplicación deberá presentar una interfaz gráfica de usuario, intuitiva y fácil de utilizar. Las clases del dominio de la aplicación deberán estar en un paquete diferente a las de la gestión de la interfaz de usuario. El código de la interfaz de usuario y de las clases del modelo debe estar estrictamente separado, de manera que las clases del modelo puedan reutilizarse en el futuro en otros contextos, por ejemplo, si se quiere hacer una versión Web o móvil de la interfaz.

**Es obligatoria la utilización de clases, estructuras de datos, persistencia y serialización en ficheros para el almacenamiento de los datos de la aplicación estando prohibido el uso de bases de datos.**

En todas las ventanas deberán visualizarse el anagrama "JavaPop" (se admiten ideas para su diseño).

Para poder comprobar el buen funcionamiento de la aplicación, se deberá poder arrancar con un mínimo de 5 usuarios y 10 productos ya dados de alta, con toda su información asociada.

Los programas fuente deberán ser fáciles de mantener por cualquier programador experto en Java. Deberán estar bien estructurados y contendrán los comentarios necesarios para un fácil seguimiento del código.

Mediante técnicas de ingeniería inversa UML deberá obtenerse el diagrama de clases de la aplicación que se incluirá en la documentación.

**Los requerimientos recogidos en el presente documento, deben entenderse como los mínimos que debe cumplir el sistema, pudiendo ser presentadas cuantas mejoras se crean convenientes.**

## Entregables para Blackboard

### Documento explicativo

La documentación de la aplicación debe redactarse para ser leída por un hipotético técnico informático que tuviese que evaluar la calidad de la aplicación desarrollada. No incluirá código fuente. Los apartados que debe incluir son:

- **Portada.** En la portada de la memoria, deberá aparecer:  
**Grado en \_\_\_\_\_". Curso 2019/20**  
**Práctica de POO 2020.**  
**Autor: DNI – Apellidos, Nombre**
- **Especificación de requisitos.** Mediante *diagramas de casos de uso* se deben describir las acciones principales que los usuarios de la aplicación pueden realizar.
- **Análisis y diseño de la aplicación.** Descripción del análisis y diseño de cómo se ha desarrollado la aplicación y qué elementos principales intervienen en la misma. Para realizarlo se construye un *diagrama de clases* sencillo, identificando las clases de la lógica de negocio.
- **Implementación (JavaDoc).** Para completar este apartado se debe haber realizado una correcta documentación de la aplicación donde se han detallado sus clases principales (clases de lógica de negocio) y su funcionalidad y las estructuras de datos utilizadas para almacenar la información que maneja la aplicación. Esto se puede realizar mediante los comentarios del código fuente para la generación de la documentación en JavaDoc.
- **Manual de usuario (descripción del funcionamiento).** En este apartado se explicará cómo interactúa un usuario con la aplicación. Es conveniente capturar las pantallas de la aplicación (se hace con las teclas Alt + Impr Pant y luego se pegan en el documento) y mostrar el funcionamiento de las distintas opciones, mensajes de error, etc.

### Proyecto NetBeans

Acompañando a la memoria del proyecto, se incluirá un proyecto NetBeans con Maven listo para arrancar la aplicación, para comprobar que se cumplen las especificaciones formuladas. Se comprimirá en un fichero indicando el primer apellido de cada miembro del grupo y el curso 19\_20. Se entrega en la sección "Trabajos" de Blackboard en el grupo correspondiente.

## Consideraciones

- La aplicación se puede hacer en grupo de dos alumnos o de manera individual.
- La evaluación de la práctica se llevará a cabo mediante una entrevista en la que el alumno deberá mostrar el funcionamiento de su aplicación y someterse a las preguntas que se consideren necesarias. Las pruebas se podrán realizar en el ordenador portátil del alumno.
- Con el objetivo de verificar la autoría de la práctica, el profesor podrá solicitar la inclusión de algún elemento de código no recogido en el presente enunciado o la modificación de algún aspecto de la aplicación. Los alumnos que no sepan introducir adecuadamente las modificaciones solicitadas serán evaluados como SUSPENSO, independientemente del contenido y calidad de la práctica entregada.

**La entrega de la práctica junto con su memoria correspondiente se realizará a través de la sección Trabajos de Blackboard. En el caso de que la práctica se realice en grupos de dos alumnos, solo es necesario que lo envíe uno, indicando los nombres de los integrantes del grupo.**

**Para facilitar el desarrollo de la aplicación se harán dos fases de entrega:**

1. **En la primera fase se entregará el proyecto NetBeans con las clases principales (clases de lógica de negocio) de la aplicación sin la interfaz gráfica, con una clase main donde se instancien objetos y se pruebe su funcionalidad básica. La fecha límite de entrega será el 10 de mayo.**

**Para la prueba de esta parte no se necesita crear un menú, con crear varios objetos es suficiente, por ejemplo, dos usuarios y cinco productos (repartidos entre los usuarios) y realizar la venta de alguno después de mostrar los resultados de una búsqueda.**

**La calificación de esta parte será el 20% de la nota del laboratorio.**

2. **En la segunda fase se entregará toda la aplicación con su documentación. Solo se calificará la entrega final de la aplicación completa. La fecha límite será el 31 de mayo. A partir de esta entrega cada profesor especificará la fecha para la defensa de la práctica.**

**La calificación de esta parte será el 80% de la nota del laboratorio.**