



UNIVERSIDAD AUTONOMA DEL ESTADO DE MEXICO
CENTRO UNIVERSITARIO UNIDAD TIANGUISTENCO

INGENIERIA EN SOFTWARE

PROGRAMACION PARALELA

MANEJO DE HILOS CON TASK EN EL SISTEMA DE COMPRAS

DOCENTE: JESUS MARES MONTES

ALUMNO: EMMANUEL MARTINEZ DIAZ

SEPTIMO SEMESTRE

Documentación del Proyecto – Manejo de Hilos con Task en el Sistema de Compras

Introducción

En el desarrollo de aplicaciones modernas, es indispensable ofrecer interfaces rápidas, dinámicas y que no se vean interrumpidas por operaciones que consumen tiempo, como consultas a bases de datos, cálculos matemáticos o generación de reportes.

En este proyecto de **tienda en línea**, se emplea **JavaFX** como framework para la interfaz gráfica y **MySQL** como gestor de base de datos. Para el manejo de concurrencia, se hace uso de la clase **Task**, que extiende de `FutureTask` y permite ejecutar procesos en segundo plano mientras la aplicación principal (JavaFX Application Thread) sigue respondiendo a las acciones del usuario.

El sistema está estructurado en **7 hilos principales**, cada uno con responsabilidades bien definidas dentro del ciclo de vida de una compra: desde que el usuario inicia sesión hasta la generación de un ticket en PDF que funciona como comprobante.

El objetivo de esta documentación es explicar detalladamente el papel de cada hilo, cómo se implementa con `Task` y qué beneficios ofrece en términos de eficiencia, seguridad y experiencia de usuario.

Desarrollo

Hilo 1 – Inicio de Sesión y Registro (10%)

Función:

El primer hilo se encarga de autenticar a los usuarios registrados o permitir que un nuevo usuario se registre en el sistema.

Implementación:

Cuando un usuario introduce su correo y contraseña, se ejecuta un `Task<Usuarios>` que consulta la base de datos y valida las credenciales. Este proceso, al ser potencialmente lento (debido a la conexión con la BD), se ejecuta en segundo plano evitando congelar la interfaz.

Beneficio:

- Permite al usuario tener una respuesta inmediata sobre la validez de sus credenciales.

- Garantiza la seguridad al validar el acceso con contraseñas encriptadas.
 - Mantiene la interfaz activa y receptiva, incluso si la base de datos tarda en responder.
-

Hilo 2 – Sesión Activa (15%)

Función:

Mantener la información del usuario activo durante toda la navegación por el sistema.

Implementación:

Se utiliza una clase auxiliar llamada **SessionManager** que guarda al usuario logueado en memoria. De esta manera, cualquier controlador puede consultar al usuario actual mediante `SessionManager.getUsuarioActual()`.

Beneficio:

- Evita tener que consultar repetidamente la base de datos.
 - Permite que los módulos de compras, carrito, historial y ticket siempre estén vinculados al usuario correcto.
 - Incrementa la cohesión del sistema, asegurando que cada acción esté asociada a una sesión válida.
-

Hilo 3 – Selección de Productos (15%)

Función:

Permitir que los usuarios seleccionen productos y los agreguen a un carrito de compras.

Implementación:

Cada vez que el usuario añade un producto, se ejecuta un `Task<Void>` que actualiza de manera segura la colección de productos en el carrito (`ObservableList<CarritoItem>`). Esta lista se sincroniza con la vista (`TableView`) de manera inmediata.

Beneficio:

- El usuario observa en tiempo real los productos añadidos al carrito.
- Se mantiene la integridad de los datos con métodos sincronizados, evitando errores de concurrencia.

- La interfaz permanece estable sin bloqueos, aunque se agreguen muchos productos de forma continua.

Hilo 4 – Cálculo de Precios (15%)

Función:

Calcular el subtotal del carrito con base en los productos seleccionados y sus cantidades.

Implementación:

Se ejecuta un Task<Double> (getTotalTask) que recorre los productos en el carrito y suma el total de forma asíncrona. El resultado se refleja en un Label que muestra el subtotal al usuario.

Beneficio:

- Se asegura que los cálculos se hagan de manera rápida y eficiente sin interrumpir la interfaz.
- Permite recalcular automáticamente los precios cada vez que se agregan o eliminan productos.
- Mejora la transparencia para el usuario al mostrarle siempre el monto actualizado.

Hilo 5 – Aplicación de Descuentos (15%)

Función:

Aplicar promociones dinámicas según las reglas de negocio definidas (por ejemplo, un 30% de descuento al superar un cierto monto o cantidad de productos).

Implementación:

Se usa un Task<Double> que evalúa el subtotal y el número de productos, y con base en estas condiciones calcula el porcentaje de descuento. Este valor se refleja en un Label diferenciado con color para que el usuario visualice claramente el beneficio aplicado.

Beneficio:

- Ofrece al cliente una experiencia atractiva al mostrar descuentos automáticos.
- Se calculan de forma precisa y en tiempo real, incentivando la compra.

- Mejora la presentación visual al destacar con colores los descuentos y el total.

Hilo 6 – Finalización de la Compra (15%)

Función:

Confirmar la compra, simular el pago y registrar la transacción en la base de datos.

Implementación:

Se implementa finalizarCompraTask(idUsuario) que realiza los siguientes pasos:

1. Calcula subtotal, descuento y total.
2. Inserta la información de la compra en la tabla **Ventas**.
3. Inserta cada producto con su cantidad y precio en la tabla **Detalle_Ventas**.
4. Limpia el carrito una vez concluida la transacción.
5. Notifica al usuario con un mensaje de confirmación en la interfaz.

Beneficio:

- Garantiza la integridad de la información almacenada en la base de datos.
- Ofrece una retroalimentación inmediata al usuario, confirmando que su compra fue exitosa.
- Aísla la lógica de negocio en un Task, evitando bloqueos en la interfaz durante la escritura en la base de datos.

Hilo 7 – Generación de Ticket en PDF (15%)

Función:

Crear un comprobante digital de la compra que pueda ser almacenado o impreso por el usuario.

Implementación:

Una vez registrada la compra, se ejecuta generarTicketTask(...). Este hilo utiliza librerías de PDF para construir un documento con:

- Fecha de compra.
- ID del usuario y ID de la venta.
- Tabla con productos, cantidades, precios y subtotales.

- Descuento aplicado y total final.

Al finalizar, el sistema abre automáticamente el archivo PDF para que el cliente pueda revisarlo.

Beneficio:

- Brinda un comprobante formal y legible al usuario.
- Permite conservar un respaldo digital de cada transacción.
- Agrega valor al sistema al simular un ticket real como el de una tienda física.

Conclusión

El desarrollo de este sistema de compras demuestra la importancia del manejo de hilos en aplicaciones con interfaz gráfica. Gracias al uso de **Task en JavaFX**, se logró un sistema donde las operaciones críticas como cálculos, consultas a base de datos y generación de archivos se ejecutan en segundo plano, manteniendo una interfaz fluida y responsiva.

Cada uno de los **siete hilos** cumple un rol clave en el flujo de compra:

- El inicio y la gestión de sesión garantizan la seguridad.
- La selección de productos y el cálculo de precios dan transparencia al proceso.
- Los descuentos y la finalización de compra hacen atractiva la experiencia del usuario.
- Finalmente, la generación del ticket en PDF otorga un comprobante real de la transacción.

En conjunto, este proyecto muestra cómo el **paralelismo y la concurrencia** son esenciales en sistemas modernos, ya que permiten mantener la **responsividad de la interfaz**, mejorar la **experiencia de usuario** y garantizar la **integridad de los datos** en procesos complejos.