

# Práctica 2: RMI

## Descripción

---

Una empresa de distribución quiere desarrollar una aplicación para gestionar el sistema de logística de pedidos entre sus *tiendas* y el *almacén* central de productos. La empresa, debido a su estrategia de concentración, sólo vende un producto. Todo el inventario de este producto es guardado en el *almacén* central que se mencionó antes. Por su parte, las *tiendas* no tienen producto almacenado localmente sino que lo solicitan al almacén central al realizar una venta del producto.

Además, para reducir costes, la contabilidad de ventas realizadas e ingresos obtenidos por las *tiendas* estarán centralizados en el *almacén* central. Por lo tanto, el *almacén* deberán llevar dicha contabilidad diferenciando por *tienda*.

La empresa tiene una política de mantener precios uniformes en todas las tiendas, por lo que el precio de cada unidad del producto vendido es de 10€.

## Funcionalidad básica

---

De acuerdo al esquema logístico de la empresa, descrito antes, se pide implementar con RMI un sistema que de soporte a los requisitos. Dicho sistema debe soportar las siguientes operaciones remotas:

- **RegistrarTienda(...)**: Con esta operación, una *tienda* se registrará en el *almacén* central, para que el *almacén* tenga constancia de la existencia de la *tienda*. La *tienda* tendrá que indicar al *almacén* su ID para que el *almacén* pueda llevar la contabilidad de ventas concretas de esa *tienda*.  
La operación devolverá si el registro se ha realizado correctamente o no.
- **RealizarVenta(...)**: Esta operación se realiza desde las *tiendas* y permite **registrar una venta realizada**. Esta operación **decrementará en 1 unidad el inventario** de producto del *almacén* central. También deberá indicar al *almacén* el ID de la tienda que realizó la venta para que se inpute correctamente a la tienda adecuada.  
La operación devolverá si la venta se ha realizado correctamente o no.  
En caso de intentar realizar una venta cuando no haya inventario, es decir, cuando el inventario del *almacén* sea 0, la operación deberá devolver una excepción *RoturaStock* para indicar dicho evento.
- **TotalizarVentas(...)**: Esta operación se realiza desde las *tiendas* y permite averiguar el **número total de unidades de producto vendidas**. La operación deberá indicar al *almacén* el ID de la tienda para retornar los valores asociados a dicha *tienda*.  
La operación devolverá número total de unidades de producto vendidas para el ID de la *tienda* proporcionado.

- **TotalizarIngresos(...)**: Esta operación se realiza desde las *tiendas* y permite averiguar **los ingresos obtenidos por las ventas** de una *tienda*. La operación deberá indicar al *almacén* el ID de la tienda para retornar los valores asociados a dicha *tienda*.  
La operación devolverá los ingresos obtenidos por las ventas asociadas al ID de la *tienda* proporcionado.

**IMPORTANTE:** notar que la información de descripción de las operaciones sólo enumera detalles de la lógica de negocio, no detalla elementos necesarios a devolver (e.g. excepciones) propios de RMI.

Por su parte, una *tienda* consistirá en un proceso con una interfaz de usuario sencilla en modo texto (la empresa no quiere interfaces gráficas). En primera instancia la *tienda* debe ofrecer al usuario la opción de registrar la *tienda*.

Una vez la tienda se ha registrado, la interfaz debe ofrecer las siguientes opciones de operaciones que se pueden realizar:

- **Vender productos:** operación para registrar en el sistema la venta de una cierta cantidad de productos. El usuario debe indicar por teclado el número de productos vendidos.
- **Calcular ventas:** operación para totalizar el número de productos vendidos.
- **Calcular ingresos:** operación para totalizar los ingresos de las ventas realizadas.

Respecto al *almacén*, debe ofrecer una interfaz de texto (por línea de comandos), para poder introducir la cantidad de producto presente en el inventario.

Se recuerda que se **debe seguir la metodología de desarrollo ágil** explicada en su documento correspondiente.

## Funcionalidad extra

---

La empresa quiere evitar que se produzcan roturas de stock, por lo que pide que el *almacen* indique (RMI-Callback) a una *tienda* que realiza una venta, si el **inventario de producto contiene 5 unidades o menos**.

Cuando el *almacén* haya notificado sobre este evento, la tienda tendrá que invocar la siguiente operación remota:

- **RealizarPedido(...)**: Esta operación se realiza desde las *tiendas* y tendrá como resultado que el *almacén* realice un pedido de 10 unidades de producto, lo que resulta en un incremento de 10 unidades en el inventario.  
La operación devolverá si el pedido se ha realizado correctamente o no.

**IMPORTANTE:** notar que la información de descripción de las operaciones sólo enumera detalles de la lógica de negocio, no detalla elementos necesarios a devolver (e.g. excepciones) propios de RMI.

## Evaluación de la práctica

---

Los pesos para la evaluación de la práctica son:

- **50%** - implementar la funcionalidad básica
- **10%** - implementar la funcionalidad extra
- **40%** - uso de la metodología

## Fecha de entrega

---

Como indica el calendario de Sprints en la 'Guía de desarrollo ágil' la fecha de entrega de la práctica será al final del Sprint 4:

**Domingo, 1 de diciembre a las 23:59.**

Posibles modificaciones en el repositorio posteriores a esa fecha no serán tenidas en cuenta, y podrían penalizar en la nota de la práctica.

## Contenido a entregar

---

El código que se corregirá de cada grupo de prácticas será el que esté en GitLab tras finalizar el periodo establecido para el desarrollo.

Además, cada grupo deberá subir al proyecto correspondiente a la práctica 1, un fichero de texto (tipo README.txt o README.md) en el que se describa brevemente lo siguiente:

- El contenido del directorio raíz de vuestra práctica, no es necesario indicar el contenido de cada subdirectorio.
- Una breve explicación de cómo ejecutar vuestra práctica (bien explicando las instrucciones de los pasos a seguir o proporcionando un fichero tipo Makefile o equivalente).
- Un listado de los componentes (clases java) que forman la práctica, indicando brevemente una descripción de las mismas.
- Qué requisitos se han desarrollado, tanto de la funcionalidad básica, como de la funcionalidad extra (si ésta última se ha realizado). Además, debéis relacionar los requisitos de funcionalidad indicados en el enunciado y que hayáis implementado, con las clases Java indicadas anteriormente.