# PRÁCTICA 7: EL DÍA DE LA MARMOTA

## 1. Introducción

Los lenguajes de programación de alto nivel, como los lenguajes de programación orientados a objetos, proporcionan una serie de primitivas y operaciones que permiten a los desarrolladores abstraerse de muchos detalles de bajo nivel. No obstante, dichos detalles de bajo nivel pueden afectar en ocasiones a ciertas propiedades de una aplicación, tales como la seguridad o el rendimiento.

Por ejemplo, la mayoría de lenguajes de programación actuales ofrecen facilidades para invocar objetos remotos como si fuesen objetos locales. No obstante, estas facilidades deben utilizarse no sin cierta cautela, pues una invocación masiva y habitual de objetos remotos podría degradar el rendimiento de la aplicación y sobrecargar la infraestructura de red innecesariamente.

Otro ejemplo clásico sería creación de objetos pertenecientes a clases cuyo proceso de instanciación es complejo. Ejemplos típicos de dichas clases podría ser apertura de conexiones a bases de datos, el acceso a servicios externos que requieran largos procesos de negociación o autenticación, o la creación de objetos de acceso a documentos XML que precisen de la carga previa de dicho documento en memoria.

Para minimizar el impacto en el rendimiento de estas operaciones existen una serie específica de patrones de diseño. Por ejemplo, para solventar el problema de la creación de objetos con complejos procesos de instanciación suelen utilizarse *pools* de recursos u objetos.

El objetivo de esta práctica es que el alumno adquiera familiaridad con este problema mediante la creación de un pool de conexiones a bases de datos. La siguiente sección resume dichos objetivos y los complementa con otros objetivos secundarios.

## 2. Objetivos

Los objetivos concretos de esta práctica son:

1. Comprender que ciertas acciones realizadas dentro de programa software, aunque puedan parecer equivalentes a otras, pueden tener un serio impacto sobre ciertas propiedades del producto software.
2. Aprender a solventar el problema de la creación de objetos complejos mediante la utilización de un *Object Pool*.
3. Aprender a solventar el problema del agotamiento de recursos dentro de un *Object Pool* mediante diversas estrategias.
4. Aprender a integrar un *Object Pool* en una aplicación real de manera adecuada mediante la aplicación del patrón *Singleton*.

## 3. Actividades

El alumno, para alcanzar el objetivo general perseguido, deberá completar las siguientes actividades, utilizando para ello el lenguaje de programación Java:

1. Crear un *object pool* de conexiones *JDBC (Java DataBase Connectivity)*, utilizando *DriverManager*s, a bases de datos. El número de conexiones que contendrá inicialmente el pool debe ser configurable. En caso de que se solicite una conexión y el *pool* no tenga en ese momento conexiones disponibles, se devolverá *null*.
2. Aplicar el patrón estrategia a la implementación anterior para permitir que el *pool*, en el caso de que no tenga conexiones disponibles, aparte de devolver *null,* pueda crear conexiones bajo demanda. Además, en esta nueva estrategia, cuando se retornen conexiones al *pool*, si éste ha alcanzado su máxima capacidad, las conexiones simplemente se cerrarán y no se incorporarán al *pool* para su reutilización.
3. Convertir el *pool* en un *Singleton*.
4. Crear un conjunto de prueba que permita verificar el correcto funcionamiento del *pool* creado.

## 4. Criterios de Evaluación y Aclaraciones

La práctica se entregará a través de la plataforma *moodle* siguiendo las instrucciones en ella proporcionadas.

La práctica se calificará atendiendo a los siguientes criterios de evaluación:

(1) Correcta creación del *pool* básico descrito en la actividad 1 (4.5 puntos).
(2) Modificación del pool para que permita la creación y destrucción de conexiones bajo demanda (2.5 puntos).
(3) Correcta aplicación del patrón *Singleton* al *pool* de conexiones (1 punto).
(4) Corrección del conjunto de pruebas creado (1 punto)
(5) Cumplimiento de la *Guía de Buenas Prácticas en Programación* (1 punto).

*Pablo Sánchez Barreiro*