

# SESIÓN 2 Patrones Factoría Abstracta, Método Factoría y Prototipo

Grupo Pequeño 6  
Pablo Rodríguez Ávila  
Miguel Medina Martínez

## Ejercicio S2E1 (Obligatorio)

Una vez diseñado el diagrama de clases, pasamos a implementarlo en Java. Nuestro proyecto se basa en un supermercado que tiene departamentos y personal. La factoría abstracta será el supermercado que tiene las funciones crearDepartamento y contratarPersonal. Por otro lado tenemos la factoría mercadona y la factoría día, que en función del supermercado los trabajadores tendrán por ejemplo un horario distinto. Todo esto se refleja en el diagrama de clases.

Para la utilización de hebras, hemos simulado un día laboral de un empleado del supermercado Día y de un empleado del supermercado Mercadona. Ambos se ejecutan de manera concurrente usando la clase Thread (la clase personal hereda de Thread). De esta manera, cada uno muestra por la interfaz de texto las horas que ya han realizado (1Hora equivale a 1 segundo). Mientras que el empleado del día trabaja 8 horas, el del mercadona trabaja 10 horas.

## Ejercicio S2E2 (Optativo)

En este ejercicio implementamos lo diseñado en el punto anterior pero cambiando varios aspectos. Se desarrolla en Ruby y en lugar del patrón factoría abstracta y el patrón método factoría, usaremos el patrón factoría abstracta con el patrón prototipo. Esto queda reflejado en el diagrama de clases. Como se puede ver tendremos una clase Factoria Abstracta que tiene relación de asociación con ambos prototipos (los prototipos son clases abstractas). Para hacer que los prototipos sean clases abstractas en ruby, hemos hecho privado los constructores (método new) y posteriormente público en las clases que heredan de los prototipos. Como se puede ver utilizamos los métodos clone para la implementación del patrón prototipo.

Aunque no es requisito la programación concurrente, la hemos implementado para aprender como se realiza en Ruby.