**Ingeniería de Software**
**Año 2025**
**Práctico 4: Patrón Factory**

**Ejercicio 0.** Descargue el código provisto en el repositorio git a continuación:

https://github.com/ingenieria-de-software-unrc/practicos-ingenieria-25

La carpeta `4-factory` contiene el código complementario a esta práctica. Se proveen scripts gradle para construir (*build*) automáticamente el proyecto (`./gradlew build`), ejecutar el main (`./gradlew run`), y ejecutar los tests (`./gradlew test`).
Se recomienda utilizar algún IDE (por ejemplo, IntelliJ IDEA) para facilitar la codificación, el refactoring, la ejecución de tests y el debugging.

**Importante:** Tenga en cuenta los siguientes requisitos al realizar los ejercicios. Piense en un diseño para su aplicación y realice un diagrama de clases del diseño propuesto antes de comenzar a codificar. Mantenga el diagrama de clases actualizado con el diseño de su aplicación durante todo el ejercicio. Asegúrese que su diseño adhiere (en la medida de lo posible) a los principios de diseño vistos en la teoría. Utilice la metodología de TDD.

**Ejercicio 1.** Utilice el código provisto de Pizza Store para experimentar y entender las distintas versiones del patrón Factory.

a) Agregue una nueva variedad argentina de pizza a la versión de PizzaStore que utiliza Simple Factory (paquete `simplefactory`).
b) Agregue una nueva versión de `PizzaStore` con variedades Argentinas de Pizzas a la implementación que utiliza Factory Method (paquete `factorymethod`)
c) Agregue una nueva familia de ingredientes para la `PizzaStore` Argentina del inciso anterior a la implementación que utiliza Abstract Factory para los ingredientes (paquete `abstractfactory`)
d) ¿Tuvo que modificar la implementación de alguna de las clases existentes, a excepción quizás de las factories, para resolver los puntos a, b y c anteriores?

**Ejercicio 2.** Transforme la versión de `PizzaStore` que usa el patrón Factory Method en una versión equivalente, pero que use el patrón Abstract Factory.

¿Cuál es la diferencia principal entre los patrones Factory Method y Abstract Factory? (piense en la forma en la que permiten extender el código).

**Ejercicio 3.** Refactorice el código del ejercicio 3 de la práctica anterior (su implementación del Conway's Game of Life) para utilizar el patrón Factory en la creación de objetos, principalmente para las partes de su implementación que cambian en las distintas versiones del juego. ¿Qué versión o qué versiones del patrón son las convenientes para utilizar en su aplicación? Asegúrese de desacoplar tanto como sea posible los componentes del juego de la creación de objetos, respetando los principios de inversión de dependencias y de programación respecto de abstracciones.