# Trabajo Practico

## INGENIERIA EN SOFTWARE 2

UADER – Facultad de Ciencia y Tecnología

**Profesor:** Pedro Colla

**Carrera:** Licenciatura en Sistemas de Información

**Alumno:** Franco Leonardo Camen.

**Github:** https://github.com/FrancoCamen/UADER\_IG2\_-Camen-.git

**Ciclo lectivo:** 2024.

# Consigna 7

Imagine una situación donde pueda ser de utilidad el patrón “abstract factory”.

Se plantea una situación hipotética donde se está desarrollando un sistema de software para una empresa de fabricación de automóviles. En este sistema, hay una parte donde los usuarios pueden personalizar y ordenar su propio automóvil. Sin embargo, la empresa ofrece diferentes líneas de productos, como vehículos económicos, vehículos deportivos y vehículos de lujo. Cada línea de productos tiene sus propias características y opciones específicas.

Aquí es donde el patrón Abstract Factory puede ser útil. Es utilizarlo para proporcionar una manera de crear familias de objetos relacionados sin especificar sus clases concretas. En este caso, podrías tener una fábrica abstracta llamada CarFactory, con métodos abstractos como createEngine(), createInterior(), createExterior(), etc. Luego, implementarías fábricas concretas como EconomyCarFactory, SportsCarFactory y LuxuryCarFactory, cada una de las cuales implementa los métodos abstractos para crear componentes específicos de sus respectivas líneas de productos.