



SIMULACIÓN DE UNA EMPRESA DE TRANSPORTE DE AUTOMOVILES.

El objetivo de esta práctica es simular el funcionamiento de una empresa de transportes, que distribuye automóviles desde el almacén de la fábrica hasta los diferentes concesionarios de una conocida marca. Cada uno de los automóviles fabricados se identifica con los siguientes datos:

- N° de Bastidor: cadena formada por 5 letras y 3 dígitos.
- Modelo: se fabrican al menos 6 modelos diferentes.
- Color: se fabrican al menos 6 colores diferentes.
- Concesionario asignado (número entero entre 1 y máximo de concesionarios NC).

Los camiones de la empresa distribuyen los automóviles entre los diferentes concesionarios que se agrupan en cuatro zonas diferentes (N, S, E, O). Se dispone de un almacén por zona con dos camiones cada una. Cada vez que la carga de un camión está completa y sale para realizar el reparto, comienza a cargarse el otro.

Para comenzar la simulación, se crearán aleatoriamente los datos de **NV** automóviles (excepto el concesionario que será 0 por defecto) que se almacenarán en una *cola_fabrica* (almacén de la fábrica).

A continuación, en cada paso de la simulación, se sacan **NS** automóviles de la *cola* (almacén de la fábrica) y se llevan a una de las zonas (elegida aleatoriamente: N, S, E, O). Cuando un automóvil sale de la fábrica hacia el almacén de la zona, se carga en el camión disponible en ese momento (*pila*).

Cuando uno de los camiones está completo (una *pila* puede cargar **NP** automóviles) se deja en el almacén de la zona correspondiente. En cada zona se utiliza un registro (*cola_zona*), que almacena los automóviles en el orden en el que se descargan del camión correspondiente. Al ir descargando los automóviles del camión y antes de ser registrados, se les asigna concesionario (de forma secuencial o aleatoria) con lo que simula el reparto a los concesionarios.

Se pide implementar, al menos, una clase **NodoPila**, una clase **NodoCola**, una clase **Pila**, una clase **Cola** (ambas con las operaciones habituales y otras que consideréis necesarias), una clase **Zona de Reparto** (que contiene dos camiones o *pilas* y un registro o *cola_zona*) y una clase **Gestor** en la que se hará la simulación.

El **programa** mostrará un **Menú con las siguientes opciones**:

1. Generar aleatoriamente la *cola de automóviles disponibles en la fábrica* con NV automóviles.
2. Generar aleatoriamente la *cola de automóviles disponibles en la fábrica* solicitando NV por pantalla.
3. Mostrar en pantalla los datos de la *cola de automóviles disponibles en la fábrica*.
4. Borrar la *cola de automóviles disponibles en la fábrica*.
5. Avanzar un paso en la simulación: salen NS automóviles de la *cola_fabrica* (almacén de la fábrica) y llegan a un almacén de zona (elegida aleatoriamente).
6. Avanzar un paso en la simulación, solicitando NS por pantalla: salen NS automóviles de la *cola_fabrica* (almacén de la fábrica) y llegan a un almacén de zona.
7. Mostrar en pantalla los datos de los almacenes de zona (camiones o *pilas*) y registro o *cola_zona*.
8. Realizar la simulación hasta finalizar los automóviles disponibles (en cada paso salen NS automóviles). En cada uno de los pasos se mostrarán en pantalla los datos de la *cola de fábrica* y de cada una de las zonas de reparto (*pilas* y *cola*).
0. Salir.



OBSERVACIONES:

- Los datos NC (entre 4 y 10), NV (entre 10 y 30), NS (entre 3 y 8) y NP (entre 7 y 12) se fijarán al inicio del programa de forma que puedan modificarse en cada ejecución si se considera necesario.
- Los modelos y colores de los automóviles fabricados serán definidos por el programador.

NORMAS PARA LA REALIZACIÓN Y ENTREGA DE LA PRÁCTICA

Las prácticas se realizarán en grupos de dos alumnos **que deberán ser los mismos para las dos prácticas** de la asignatura. Deberán **confirmarse** las personas que forman el grupo **antes del día 18 de octubre** (durante la clase o email al profesor correspondiente).

1. La práctica se implementará en C++, utilizando CodeBlocks. Debe entregarse un fichero comprimido incluyendo **todos los ficheros fuente del proyecto** C++ y el documento descrito en el punto 4. Se subirá un fichero por grupo a la plataforma **antes de la fecha indicada para cada una de las prácticas**. El nombre del fichero será el nombreapellido1apellido2 de uno de los miembros del grupo.
2. En la **defensa** de la práctica **se verificará la autoría de la práctica entregada y será calificada con APTO/NO APTO, siendo necesaria la calificación de APTO para poder ser evaluado de la práctica**.
3. La entrega de prácticas copiadas supondrá el suspenso de la asignatura en esta convocatoria para todos los alumnos implicados.
4. La documentación que se subirá en un fichero *.pdf* junto con el proyecto (tendrá un peso del 10% de la nota), deberá tener al menos los siguientes apartados:
 - a. Nombre y DNI de los alumnos del grupo.
 - b. Detalles y justificación de la implementación:
 - i. b.1 Especificación concreta de la interfaz de los TAD's implementados:
 - j. b.1.1 TAD's creados.
 - j. b.1.2 Definición de las operaciones del TAD (Nombre, argumentos y retorno).
 - j. b.2 Solución adoptada: descripción de las dificultades encontradas.
 - j. b.3 Diseño de la relación entre las clases de los TAD implementados.
 - k. b.3.1 Explicación de los métodos más destacados.
 - k. b.3.2 Explicación del comportamiento del programa.
 - k. b.4 Bibliografía.



PRÁCTICA 1: FECHAS E INDICACIONES

1. La práctica 1 se entregará antes del **12 de noviembre a las 23:59 horas**. Tal y como se describe en el enunciado, es elección del programador la forma de crear aleatoriamente los datos de los clientes y las entradas. En esta primera práctica, se entregarán las opciones desde la 1 hasta la 8.
2. La **defensa**, individual y obligatoria para ser calificado, será durante la **clase del 15 de noviembre** (si es necesario se fijará otra sesión/horario para finalizar las defensas) y constará de dos partes:
 - 2.1 Defensa escrita (se realiza sin máquina) se puede traer, si se desea, el código en papel.
 - 2.2 Defensa oral: para su realización es necesario traer el código entregado en el portátil o en un pen drive (en caso de no traer el portátil).

EN CASO DE NO PRESENTARSE A LA DEFENSA NO SE CALIFICARÁ LA ENTREGA.