



ARQUITECTURA DE COMPUTADORES

Grado en Ingeniería Informática

**Práctica – Programación
Orientada a Objetos:
“Detección de vehículos”**

**Dr. D. Miguel Ángel Conde González
David Martínez Martínez**

Índice

1. Introducción	3
2. Consideraciones	5
3. Formato de entrega.....	5

1. Introducción

Desarrollar una aplicación gráfica utilizando el lenguaje C++ que permita configurar vehículos, almacene dicha información en una clase y decida el resultado final.

La aplicación debe permitir la configuración de los siguientes elementos:

- Nombre del vehículo.
- Número de ruedas: Se podrá elegir entre 2, 3, 4, 6 y 40.
- Motor: Sí o no.
- Potencia del motor: El motor deberá tener entre 80 y 450 cv. La potencia del motor solo se podrá elegir si el vehículo tiene motor.
- Combustible: Con o sin combustible.
- Tipo de combustible: Eléctrico, gasolina, diésel, híbrido o queroseno. El tipo de combustible solo se podrá elegir si el vehículo tiene combustible.
- Color: Blanco, Negro, Rojo, Azul, Amarillo y Verde.
- Alas: Sí o no.
- Reactores: Sí o no.
- Tren de aterrizaje: Sí o no.
- Locomotora: Sí o no.
- Vagones: Podrá tener 0 vagones o entre 5 y 20 vagones.
- Rueda de repuesto o Kit de reparación de pinchazos.
- Matrícula: La matrícula debe generarse de forma aleatoria pulsando un botón, siguiendo el patrón de 4 números (desde el 0000-9999) seguido de 3 letras. No se pueden utilizar vocales en las matrículas.

La aplicación deberá decidir si el vehículo se trata de: una bicicleta, un triciclo, una motocicleta, un coche, un coche deportivo, un tren o un avión en función de la configuración elegida. Las condiciones de cada vehículo son las siguientes:

- **Bicicleta:** 2 ruedas, sin motor, sin combustible, de cualquier color, sin alas, sin reactores, sin tren de aterrizaje, sin locomotora, sin vagones y con kit de reparación de pinchazos.
- **Triciclo:** 3 ruedas, sin motor, sin combustible, de cualquier color, sin alas, sin reactores, sin tren de aterrizaje, sin locomotora, sin vagones y con kit de reparación de pinchazos.
- **Motocicleta:** 2 ruedas, con motor de entre 80 y 450 cv, con combustible de tipo Eléctrico, gasolina, diésel o híbrido, de cualquier color, sin alas, sin reactores, sin tren de aterrizaje, sin locomotora, sin vagones y con kit de reparación de pinchazos.
- **Coche:** 4 ruedas, con motor de entre 80 y 250 cv, con combustible de tipo Eléctrico, gasolina, diésel o híbrido, de cualquier color, sin alas, sin reactores, sin tren de aterrizaje, sin locomotora, sin vagones y con rueda de repuesto.
- **Coche deportivo:** 4 ruedas, con motor de entre 250 y 450 cv, con combustible de tipo Eléctrico o gasolina, de cualquier color, sin alas, sin reactores, sin tren de aterrizaje, sin locomotora, sin vagones y con kit de reparación de pinchazos.
- **Avión:** 6 ruedas, con motor de 450 cv, con combustible de tipo queroseno, de color blanco, con alas, con reactores, con tren de aterrizaje, sin locomotora, sin vagones y con rueda de repuesto.
- **Tren:** 40 ruedas, con motor de 450 cv, con combustible de tipo Eléctrico o Diesel, de color negro, sin alas, sin reactores, sin tren de aterrizaje, con locomotora, con vagones entre 5 y 20 y con rueda de repuesto.
- Con cualquier otra configuración la aplicación mostrará una alerta que nos indicará que la configuración es incorrecta y por tanto no se guardará el vehículo creado.

La aplicación dispondrá de un elemento gráfico que muestre el número de vehículos creados, y una vez configurado un vehículo, mostrará el tipo de vehículo creado y lo almacenará en memoria sumando uno al número de vehículos creados. Una vez creado el vehículo, la interfaz se reseteará y desde ahí se podrá:

- Crear un vehículo nuevo.

- Recuperar los datos de cualquier coche creado anteriormente.

2. Consideraciones

- Se deben utilizar varios elementos gráficos: etiquetas, cuadros de texto, botones, check box, radio button, listas desplegables, etc.
- Se deben utilizar clases para almacenar la información.
- En la medida de lo posible, se deberá separar la interfaz gráfica de la lógica del sistema.

3. Formato de entrega

- La práctica es individual.
- Esta entrega deberá realizarse a través de agora.unileon.es
- El código fuente se entregará en un único archivo comprimido.
- **Fecha límite de entrega:** 16 de marzo de 2020.