



## Examen – Plataforma: EcoLogistics Tracker

**ACLARACIONES:** Los datos deben definirse dentro del propio archivo PHP, puedes usar los proporcionados o te los puedes inventar. No se permite el uso de *bases de datos* ni *DateTime*. No se permite el uso de internet más allá del manual de PHP o la Moodle para acceder a la tarea correspondiente. Las pantallas deberán grabarse desde el inicio de la prueba. No se corregirá ninguna prueba que no incluya el correspondiente vídeo en el que se muestre la realización completa del examen. La **no entrega del vídeo** junto con la prueba o la detección de cualquier actividad sospechosa supondrá la calificación de **suspenso automático**. Se valorará, además de la funcionalidad del programa, la calidad del código, claridad, legibilidad, indentaciones, uso de nombres apropiados, modularidad, limpieza del código, organización general y buenas prácticas de programación.

La empresa **EcoLogistics** gestiona rutas de reparto ecológicas realizadas por distintos conductores y dicha empresa quiere un sistema en PHP que permita analizar rutas, conductores y vehículos.

Cada ruta incluye:

- Conductor
- Vehículo utilizado
- Fecha
- Hora de salida
- Duración (minutos)
- Distancia (km)
- Carga transportada (%)
- Zonas visitadas (array)
- Tipo de ruta (solo un tipo: *Urbana*)

### APARTADOS (10 puntos)

#### 1. Estructura y jerarquía (1,7 puntos)

Se debe crear la estructura necesaria para representar rutas y vehículos. Deben incluir:

##### Interfaz – Analizable (0.2 puntos)

- Métodos: `calcularEficiencia()` y `nivelImpacto()`. Estos métodos realizan los siguientes cálculos:



### **Cálculo de eficiencia**

- Fórmula:  $\text{eficiencia} = (\text{distancia} / \text{duración}) \times 100$
- Si la ruta pasa por el centro, la eficiencia se reduce en un 10%.

### **Cálculo del impacto**

- Impacto base: 2 puntos
- Si la carga > 70% → +2 puntos
- Si la distancia > 15 km → +2 puntos

### **Clase abstracta – Ruta (0.7 puntos)**

A tener en cuenta:

- Atributos: conductor, vehiculo, fecha, hora, duracion, distancia, carga, zonas
- Validación de conductor mediante expresión regular: solo letras y espacios, mínimo 3 caracteres.
- Validación de fecha y hora sin usar DateTime: la fecha se almacena como AAAA-MM-DD pero en la salida debe mostrarse en formato DD/MM/AAAA
- Atributo estático para contar rutas
- Getters y setters
  - Si un dato no cumple la expresión regular, se asigna un valor por defecto.
- La salida se debe mostrar en el siguiente formato:

Conductor – Vehículo – DD/MM/AAAA HH:MM

### **Clase hija – RutaUrbana (0.4 puntos)**

Como mínimo tiene:

- Añadir un atributo propio (por ejemplo: si pasa por el centro)

### **Clase Vehículo (0.4 puntos)**

- Atributos: marca, modelo, matrícula (validada con RegEx)
- Validación de matrícula: debe cumplir que contenga 4 números seguidos, un espacio y 3 letras mayúsculas. Ejemplo válido: 1234 ABC. Si no es válida, asignar "0000 XXX".

## **2. Visualización general de todas las rutas (0.8 puntos)**

Debe mostrarse una tabla con:

- Conductor
- Vehículo
- Fecha (DD/MM/AAAA)



- Hora
- Duración
- Distancia
- Carga
- Eficiencia
- Impacto

Debajo de la tabla, mostrar también:

- Total de rutas
- Total de conductores distintos

### 3. Ruta más eficiente (0.8 puntos)

Localiza la ruta con mayor eficiencia y muestra todos sus datos.

### 4. Resumen general (1,3 puntos)

Debe mostrar:

- Número total de rutas
- Distancia total
- Carga total
- Eficiencia media (con 2 decimales) **(0.5 puntos)**
- Velocidad media total del conjunto de rutas (con 2 decimales) **(0.5 puntos)**

### 5. Búsqueda de rutas por conductor (1.2 puntos)

Crear un formulario con un campo de texto. Al enviarlo, debe:

- Mostrar las rutas realizadas por ese conductor
- En caso de no tener rutas, mostrar un mensaje adecuado

### 6. Análisis de rutas por fechas (1 punto)

Sin usar DateTime:

- Mostrar cuántas rutas hay por fecha
- Indicar la fecha con mayor número de rutas

### 7. Rutas de alto impacto (1,5 puntos)

Considerar un “alto impacto” cuando  $\text{nivelImpacto} \geq 7$ .

Debe mostrarse:

1. Una lista de rutas de alto impacto



2. El número total de rutas de alto impacto
3. La ruta de alto impacto con mayor distancia recorrida

#### **8. Adjuntar archivo y análisis de zonas (1,7 puntos)**

Crear un formulario que permita:

- Subir un archivo .txt. (**0.3 puntos**)
- Introducir una lista de zonas separadas por comas (Ej: Centro, Norte, Polígono, Puerto, ...). (**1.4 puntos**)

Cuando se envíe el formulario, debe:

1. Mostrar cuántas rutas visitan alguna de esas zonas
2. Mostrar una tabla con las rutas coincidentes