

Universidad de Costa Rica
Facultad de Ingeniería
Escuela de Ciencias de la Computación e Informática
CI-0112

Tarea Programada 0

Profesor : Nombre Profesor

Estudiantes :

Carné : Nombre Completo

Carné : Nombre Completo

Carné : Nombre Completo

Índice

1. Enunciado	2
2. Diseño de la Solución	3
2.1. Diagrama de Clases	3
3. Clases	4
3.1. Carro	4
3.1.1. Atributos	4
3.1.2. Algoritmos	4
4. Manual de usuario	5
4.1. Compilación	5
5. Casos de Prueba	5
6. Conclusiones	6
6.1. Distribución de Trabajo	6

1. Enunciado

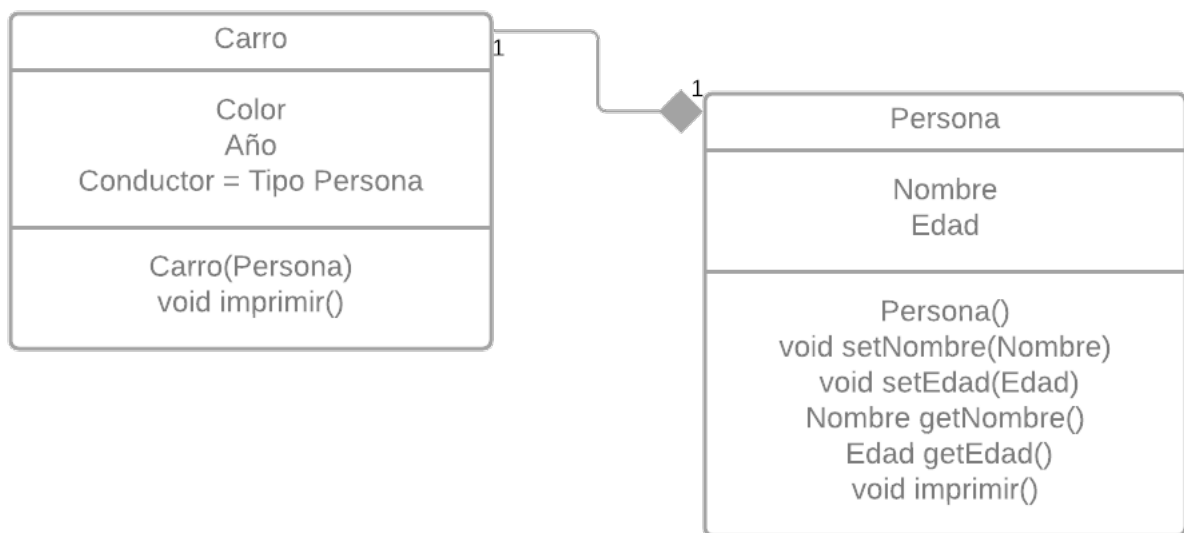
En esta parte la intención es mostrar el enunciado del problema. Se puede copiar y pegar del documento entregado por el profesor.

2. Diseño de la Solución

En una serie de párrafos, los estudiantes deben de explicar a *grosso modo* cómo se plantea la solución del enunciado. Esto es con el motivo de que haya un planeamiento previo al desarrollo de la solución.

2.1. Diagrama de Clases

El modelado es una herramienta muy útil para plantear un diseño en un programa que está diseñado bajo el paradigma de orientación de objetos. Los estudiantes deben plantear un diagrama de clases. Este diagrama **no necesariamente tiene que representar la solución final**, si no que se utiliza más como una herramienta para plantear un diseño efectivo.



Existen muchos sitios web para generar diagramas. Uno de ellos es *draw.io* o *lucidchart.com*

3. Clases

Al ser **Java** un lenguaje orientado a objetos, una buena práctica de documentación es enunciar cada clase con sus componentes.

3.1. Carro

3.1.1. Atributos

En la sección de atributos, basta con enunciar cuáles son los atributos que utiliza la clase respectiva y el motivo por el que están.

```
1 private static int BLANCO = 0;
2 private static int AZUL = 1;
3 private static int ROJO = 2;
4 private static int NEGRO = 3;
```

- Por ejemplo, este grupo de variables son enteros donde cada número representa un color diferente. Son variables de tipo `static int` para que no se pueda modificar su representación.

```
1 private int color;
```

- **color** es un atributo de la clase que va a identificar qué color posee el carro. Puede tener los valores de 0, 1, 2, 3.

3.1.2. Algoritmos

En el apartado de algoritmos, basta con explicar todos los algoritmos complejos que implementaron en la clase. Se deben mencionar las funciones o métodos que utiliza el algoritmo para funcionar correctamente.

4. Manual de usuario

En esta sección se describen los pasos de cómo utilizar el programa: si le piden datos al usuario, se debe describir qué tipo de información el usuario debe ingresar. Además, si el programa consiste en un juego o alguna otra dinámica, se debe crear el manual para que el usuario entienda lo que se puede encontrar al ejecutar el programa.

4.1. Compilación

En este apartado se debe indicar cómo compilar el programa y cómo ejecutarlo. Ejemplo: ir a la carpeta X y, en una terminal, ejecutar el comando `javac...`”

5. Casos de Prueba

En los casos de prueba se pueden mostrar pantallazos de la terminal o Interfaz que se esté utilizando.

```
-----
Informacion Carro :
• Año   : 28
• Color : Blanco
• Conductor : Nombre : Camila
Edad : 20
-----
Informacion Carro :
• Año   : 6
• Color : Blanco
• Conductor : Nombre : Eddy
Edad : 37
-----
Informacion Carro :
• Año   : 4
• Color : Blanco
• Conductor : Nombre : Ximena
Edad : 47
```

6. Conclusiones

En el apartado de conclusiones, se puede escribir en un párrafo pequeño, lo que se aprendió, los desafíos y, si es el caso, lo que no se pudo completar para el desarrollo de la Tarea Programada.

6.1. Distribución de Trabajo

Para finalizar, en esta ultima sección, se debe indicar el porcentaje de trabajo de cada miembro del grupo en la elaboración de esta tarea (programación y documentación). Ejemplo: si son dos integrantes y trabajaron la misma cantidad, indicar el nombre y un 50 % para los dos integrantes.