

	Primer Entregable	Código: FO-DOC-12
		Versión: 01
		Fecha de Aprobación: Mayo 02 de 2018
		Página 1 de 3

COPIA CONTROLADA

FACULTAD/DEPENDENCIA: Ingeniería		FECHA DE ENTREGA		
PROGRAMA ACADÉMICO/ ÁREA:		DD 11	MM 09	AAAA 2022
ASIGNATURA: Construcción de software IV		PROFESOR: Juan Guillermo Duque Galvis		
TIPO DE EVALUACIÓN	Práctica	PORCENTAJE: 15%	CALIFICACIÓN:	
ALUMNOS:(Nombres Apellido - Identificación) : Maximo 3 personas				
Juan Fernando Muñoz Urrego – 1.036.604.143				

Entrega: Realizar los ejercicios propuestos y subirlos a la plataforma [GitHub.com](https://github.com)
Además enviar este documento con el enlace del repositorio y el nombre de los integrantes al siguiente correo jduqueg2@correo.tdea.edu.co.

Ruta del GIT:

<https://github.com/jfmedellin/CIV/tree/main/Entregable2/Monoplaza-Pastel>

Lenguaje: C#

Punto 1:

Se necesita un programa que permita evaluar qué monoplaza en un circuito es más rápido entre McLaren, Ferrari o un Red Bull.

Se sabe que los vehículos de tipo monoplaza siempre tienen las funcionalidades de encender, apagar, detener y en movimiento.

Encender, solo se puede encender si el vehículo se encuentra apagado. Y además debe mostrar un mensaje indicando que está encendido.

Apagar, solo se puede apagar si el vehículo se encuentra encendido y está detenido. Y además debe mostrar un mensaje indicando que está apagado.

Detener: solo se puede detener si el vehículo está encendido y en movimiento Y además debe mostrar un mensaje indicando que está detenido.

Movimiento: solo se puede mover si el vehículo está encendido y detenido. Y además debe mostrar un mensaje indicando que está en movimiento

Los monoplazas aparte de esas acciones tienen la propiedad de escudería, la cual se asigna en el momento de crearlo

	Primer Entregable	Código: FO-DOC-12
		Versión: 01
		Fecha de Aprobación: Mayo 02 de 2018
		Página 2 de 3

COPIA CONTROLADA

El circuito va aceptar solo vehículos de tipo monoplaza. En el circuito se van a realizar las acciones de agregar monoplaza, realizar prueba y sacar monoplaza. En el circuito solo puede estar un monoplaza a la vez.

Agregar monoplaza: Este solo va asignar un monoplaza al circuito siempre y cuando no haya un monoplaza en el circuito y no emite ningún mensaje.

Sacar monoplaza: Este solo va sacar el monoplaza del circuito siempre y cuando haya un monoplaza en el circuito y no emite ningún mensaje.

Realizar prueba: Esta acción es donde el monoplaza dará **N** cantidades de vueltas al circuito, La forma de calcular el tiempo será en un número aleatorio de 6 cifras, en cada vuelta se debe imprimir el tiempo de vuelta y al final se entregará el mejor tiempo.

Cuando se crea el circuito se asignan el número de vueltas permitida y el nombre del circuito.

Notas

1. Todos los monoplazas empiezan apagados antes de realizar la prueba y deben terminar apagados al finalizarla.
2. Cuando estén dando las vueltas por ende deben estar en movimiento.

Al final el programa debe entregar la tabla de posiciones menor a mayor tiempo.

Punto 2:

Se necesita un programa que indique cuántos y cuáles ingredientes tiene el paste, como se llama el pastel y cuánto fue el costo del pastel.

Ingrediente: Tiene las propiedades de nombre, cantidad y precio, estas propiedades se ingresan cuando se construye el ingrediente.

Pastel: Tiene las propiedades de nombre, tamaño y lista de ingredientes. Y además las acciones de agregar ingrediente, cantidad de ingredientes, listar ingredientes y calcular costo

Calcular costo: retorna el valor total del pastel a partir del precio de los ingredientes.

Lista de ingredientes: retorna una cadena de texto de los ingredientes con saltos de línea.

	Primer Entregable	Código: FO-DOC-12
		Versión: 01
		Fecha de Aprobación: Mayo 02 de 2018
		Página 3 de 3

COPIA CONTROLADA

Cantidad de ingredientes: retorna la cantidad de ingredientes.

Agregar ingrediente: añade un ingrediente a la lista de ingredientes y no retorna nada

Nota: En este ejercicio no es necesario crear clases abstractas sólo la interfaz de pastel.

Ayuda con lista de ingredientes:

```
//Importamos el nombre de espacio que contiene List
using System.Collections.Generic;

//Creamos la clase ingrediente
Ingrediente Ingrediente_1 = new Ingrediente();
Ingrediente Ingrediente_2 = new Ingrediente();

//Inicializamos la lista de ingredientes
List<Ingrediente> ListaIngredientes = new List<Ingrediente>();

//Agregando un ingrediente a la lista
ListaIngredientes.Add(Ingrediente_1);
ListaIngredientes.Add(Ingrediente_2);

//Eliminar elemento
ListaIngredientes.Remove(Ingrediente_2);

//Devuelve cuantos items tiene la lista
int count = ListaIngredientes.Count;

//Recorrer la lista, cada item es una instancia de la clase Ingrediente
for(int i = 0; i < ListaIngredientes.Count; i++) {
    Console.WriteLine(ListaIngredientes[i]);
}

foreach (Ingrediente ingrediente in ListaIngredientes) {
    Console.WriteLine(ingrediente);
}
```