

## Tecnología y paradigmas de programación: Práctica 1

Crear una solución, en la que se incluirá una Aplicación de consola y una Biblioteca de Clases (Simuladores).

En la **Biblioteca de clases**, crear la **clase SimuladorTiempo**. Decide cómo implementar:

- **Fecha** (tipo DateTime). Se establece una única vez al instanciar con el valor System.DateTime.Now
- **Estación** (tipo Estaciones). Indica la estación actual (consultar fichero enums.txt)

Y, como mínimo, los siguientes **métodos**:

- **AvanzarDias**: sumará a la propiedad Fecha un valor aleatorio de [2,12) días.
- **AvanzarMeses**: sumará a la propiedad Fecha un valor aleatorio de [1,4) meses.
- **CalcularEstacion**: modificará la propiedad Estación en función del mes de Fecha: [2,4] Primavera; [5, 7] Verano; [8, 10] Otoño. En otro caso, Invierno. Este método se invocará cada vez que se modifique Fecha.
- **ToString** para que devuelva Fecha y Estación.

En la **Aplicación de Consola**, crear la **clase Cultivo**. Decide la mejor forma de implementar:

- **Cosecha** (tipo Estaciones). No modificable una vez inicializada. Estación en la que se puede cosechar.
- **RitmoCrecimiento** (tipo int) indica el ritmo de crecimiento de un cultivo. No modificable una vez inicializada.
- **Crecimiento** (tipo int). Estado del crecimiento del cultivo. No modificable desde el exterior de la clase.
- **Cantidad** (tipo int). Cantidad de semillas que se plantan. No modificable una vez inicializada.
- **Muerto** (tipo bool). Un cultivo pasa a muerto cuando su Crecimiento es < 0.
- **Identificador** (tipo string). Un identificador del cultivo.

Y, como mínimo los siguientes métodos:

- **Regar** (recibe la Estación actual de un SimuladorTiempo). Incrementa Crecimiento en base a RitmoCrecimiento. Un cultivo Muerto no se puede Regar.
- **Cosechar** (recibe la Estación actual de un SimuladorTiempo):
  - Un cultivo Muerto no se puede cosechar.
  - Si Estación recibida es igual que Cosecha: devolver un número en (cantidad\*2, cantidad\*5].
  - Si Estación no es igual que Cosecha: devolver un número en (cantidad, cantidad\*2).
  - Debe restarse a Crecimiento un valor (0, crecimiento\*2].
- **ToString** para que muestre toda la información del cultivo.

Idea una forma para, simulando el paso de los días, regar varias veces un cultivo, cosecharlo y repetir el proceso hasta que el cultivo esté Muerto.

Para generar aleatorios:

```
Random random = new Random();  
int aleatorio = random.Next(min, max);
```

Para sumar tiempo a fechas:

```
fecha = fecha.AddMonths(aleatorio); // AddDays
```

**Completar y subir la actividad obligatoria antes del inicio de la próxima sesión.**