



## Indizadores

### Ejercicio 1

---

- Crea una clase **Jugador** que permita guardar el nombre de un jugador y la puntuación obtenida en las últimas 10 partidas.
- Crea un indizador para la puntuación y un método estático **PonerPuntuacion** en la clase programa que te permita generar la puntuación del jugador aleatoriamente, utilizando el indizador.
- Si necesitas saber la longitud del array, crea una propiedad llamada Longitud (esta es otra perspectiva de cómo ver las propiedades).
- El constructor de la clase se encargará de guardar el nombre del jugador, y redefiniremos el método ToString en la clase Jugador para mostrar sus datos (nombre, puntuación).

### Ejercicio 2

---

- Utilizando las clases de **Libro** y **Autor** que hiciste en ejercicios anteriores. Vamos a modificarlas de forma que ahora cada autor poseerá un array de libros donde se guardará la información de estos.
- Para eso modifica los constructores, crea un indizador para el array libros, crea las propiedades necesarias, sobrescribe el método ToString para que te devuelva toda la información de un autor como cadena y, por último, crea un método en la clase program que te permita usar el indizador para meter información de los libros de un autor.

### Ejercicio 3

---

- A partir del ejercicio 1 deberás crearte una clase Partidas que tendrá el nombre del juego, la descripción y una array de tipo Jugador.
- El método PonerPuntuacion seguirá siendo estático pero lo pasaremos a la nueva clase.
- Además, implementaremos un método más **LlenaArrayPuntuaciones** que generará las puntuaciones de todos los jugadores usando PonerPuntuacion.
- Método **ToString** que generará la información de la partida y de los jugadores llamando al Mostrar de ToString.
- Crea un programa principal y los métodos necesarios para su correcto funcionamiento y prueba.

### Ejercicio 4

---

- Necesitamos una clase para almacenar los datos de una factura.
- **Datos factura:** Un **cliente** con nombre, teléfono, dirección, población, provincia, código postal, NIF o CIF y porcentaje de IVA.
- Por otra parte, tienes que tener presente que en una misma factura puede haber una o varias **líneas de detalle**.
- **Datos línea de detalle:** Cantidad, descripción, precio unitario e importe.
- Usa un indizador para acceder a cada una de estas líneas de detalle.
- La clase factura, además, debe ofrecer **propiedades** que devuelvan la base imponible, la cuota de IVA y el total a pagar.



- Escribid también un método **principal** cliente de esta clase que demuestre que funciona correctamente.
- Supongo que ya habrás deducido que para que la clase Factura cumpla los requisitos que te pido tendrás que construir también una clase Detalle.
- Pues bien, te propongo también que sobrecargues el operador + para que puedas sumar objetos de la clase Detalle a objetos de la clase Factura.  
Ojo, en este caso solamente queremos hacer posible la suma Factura + Detalle, nada más.

## Ejercicio 5

---

- Vamos a encapsular en una clase denominada **Presupuestos** una matriz cruzada de presupuestos donde las **filas** sean los conceptos las cuales podrán ser: Comida, Casa y Entretenimiento (enumeración).
- Las columnas se corresponderán con los 12 meses del año (enumeración). De tal manera que el contenido de una celda me indicará el presupuesto de un concepto para un determinado mes.
- Inicializaremos la matriz de tal manera que en cada mes para comida tendremos un presupuesto de 300 € para la casa de 500 € y para entretenimiento de 200 €.  
Esto lo haremos usando un for para recorrer la enumeración meses.
- Deberemos, además, crear un indizador para la clase Presupuestos que me devuelva el presupuesto para un objeto de esta clase indexado por dos enteros, dos valores enumerados de concepto y mes, y dos cadenas indicando también el concepto y el mes. Además, la clase presupuesto redefinirá el método ToString() para devolverme una cadena que me muestre mis presupuestos por meses en un primer nivel y en un segundo nivel por conceptos.
- Usa la implementación de estos indizadores para la inicialización en el Constructor (utilizando this) y el recorrido para visualización en el método ToString();

## Ejercicio 6      Ampliación

---

- Deberemos implementar una clase **Horario** que implemente un indizador que reciba un objeto **Hora** que irá de las 8:00 a las 13:00 horas y un valor **enumerado** con un **día de la semana** de Lunes a Viernes.
- La clase Hora se creará a partir de una cadena del tipo "HH:MM "e internamente almacenará los atributos hora y minutos.
- Ambos datos se le solicitarán al usuario por teclado para mostrar una actividad usando el indizador.
- **Ejemplo:**

```
string actividad = horario[new Hora("9:45"), Dia.Lunes];
```

- También redefiniremos el método ToString para mostrar todo un horario por pantalla.
- El horario lo inicializaré en un constructor por defecto con dicho indizador.