

## Ejercicios de repaso

Joan Gerard Camarena Estruch

Accés a Dades  
Ejercicios



## 1. Básicos.

Crear las siguientes aplicaciones básicas.

- Hola Mundo
- Hola Mundo desde clase `Inicio` con método `arrancar()`
- Desde clase `Inicio` sacar texto por pantalla pasado por *parámetro*. La clase `Inicio` también tendrá una propiedad `texto` que guardará el valor pasado como parametro: `arrancar(String texto)`

## 2. Arrays/Vectores

- Aplicación java que genere 100 números enteros aleatorios diferentes entre el 0 y el 1000. Que los muestre por pantalla. Tendremos una clase llamada `VectorNumeros` con los métodos necesarios para ir añadiendo los números al vector.
- Sobre el ejercicio anterior. Despues de generar los números pediremos al usuario un número por el teclado. El programa buscará el número y nos devolverá la posición en el vector del número que sea igual o más cercano. A la case `VectorNumeros` del ejercicio anterior le añadiremos los métodos necesarios para buscar un número.

Ejemplo:

```
1  /* devolvera false si el número ya existía en el vector y por tanto no
   lo puede añadir */
2  public boolean add(int n) {
3      /** Implementar **/
4  }
```

para el b. Ejemplo:

```
1  /** devolverá la posición en el vector del número más cercano **/
2  public int find(int n){
3      /* implementar */
4  }
```

Notas:

- Se deja como opcional que los números se guarden ordenados o no, para optimizar la velocidad de las búsquedas
- Otros métodos pueden ser por ejemplo 'get' para que devuelva el valor de una posición del vector etc...

- Se deja como opcional el uso de un `array` o de un `ArrayList`

### 3. Matrices. Guerra de barcos.

Crear una clase `Tablero`, debe contener una matriz de 10x10 que simboliza un tablero.

Al inicializarse la clase (constructor) debe crearse la matriz y alojar aleatoriamente 5 barcos: 2 de 2x1, 1 de 3x1, 1 de 4x1 y 1 de 5x1. Dos barcos no pueden ocupar la misma posición y deben estar completamente dentro del tablero. Tampoco pueden estar adyacentes, es decir, que alrededor de cada barco debe haber agua o nada, pero nunca otro barco. Los barcos estarán en horizontal o vertical (aleatoriamente) pero nunca en diagonal.

La clase `Tablero` tendrá todos los métodos necesarios para controlar la posición de los barcos, su estado etc..

También tendrá un método `bomba(fila, columna)` que devolverá:

- 0 si *agua*
- 1 si *tocado*
- 2 si *hundido*
- -1 si la fila o la columna *exceden* del tablero.

Reglas:

- Los barcos se posicionan de manera aleatoria
- El jugador irá dando posiciones por teclado (A1, B8 ...) como en el típico juego. donde la letra es la fila y el número la columna.
- El programa responderá al usuario *agua*, *tocado* o *hundido*.
- El juego acaba cuando el jugador se rinde, mediante el input 00 ('cero cero') o cuando se hunden todos los barcos.
- Al finalizar el programa siempre indicará el número de jugadas realizadas por el jugador.