

Tema 7 – S4

Contenidos:

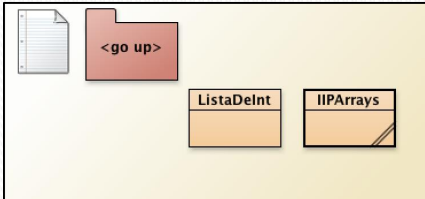
2. **Representación y tratamiento de datos mediante un array**
 - Representación de datos utilizando arrays y operaciones básicas
 - Recorrido de un array: esquemas, ejemplos y ejercicios
 - Búsqueda de un dato en un array: esquemas, ejemplos y ejercicios

Recorrido de un array: ejercicios resueltos



BlueJ: ejercicios - Tema 7

Ejercicio Nº 6 Transparencias



Completa los métodos **minimo** y **maximo** de la clase **ListaDeInt**.
Indica los cambios obligados que debes realizar para ello en el método que figura a continuación, de la clase **IIPArrays**

```
public static double minimo(double[] v) {  
    double min = v[0];  
    for (int i = 1; i < v.length; i++) {  
        if (v[i] < min) { min = v[i]; }  
    }  
    return min;  
}
```

// RECUERDA que la clase **ListaDeInt** tiene 2 atributos: **int[] elArray** y **talla**

```
public static double int minimo(double[] v) {  
    double int min = velArray[0];  
    for (int i = 1; i < v.length talla; i++) {  
        if (velArray[i] < min) { min = velArray[i]; }  
    }  
    return min;  
}
```



BlueJ: ejercicios - Tema 7

Ejercicio Nº 3 Transparencias:

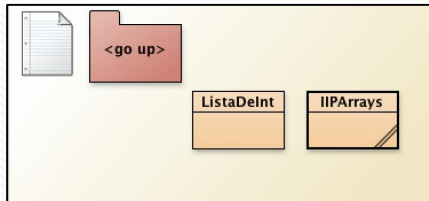
Implementa los métodos **media**, **toString** y **toArray** de la clase **ListaDeInt**

Búsqueda de un dato en un array

Análisis de los casos significativos



BlueJ: ejercicios - Tema 7



Diseña el método **buscar** en la clase de Utilidades IIPArrays, que devuelve la posición de la primera aparición de b en el array v, o -1 si b no está en v

```
public static int buscar(int[] v, int b) {  
    int res = -1; // RECUERDA: inicio = 0, fin = v.length - 1  
    // cálculo de res  
    ¿Recorrido? Asc. de v, hasta talla = v.length y cuerpo ¿v[i] == b?  
    return res;  
}
```

Analicemos los **casos significativos** para un array v concreto...
o **instancias significativas**



Búsqueda de un dato en un array

La estrategia y su traducción DIRECTA a bucle Java

Bucle de Recorrido ¡ESPECIAL!

Si b **NO** está en el array v...

➤ Recorrido **completo** de v
Búsqueda **SIN** éxito

➤ **Peor** de los casos

Si b **SÍ** está en el array v...

Búsqueda **CON** éxito → Recorrido **In**completo de v

↳ Si b **es el 1^{ero} de** v ➤ **NO hay** Recorrido
➤ **Mejor** de los casos

```
public static int buscar(int[] v, int b) {  
    for (int i = 0; i < v.length; i++) {  
        if (v[i] == b) { return i; } // Búsqueda CON éxito  
    }  
    return -1; // Búsqueda SIN éxito (caso Peor)  
}
```

Búsqueda de un dato en un array

Esquema 1: traducción **DIRECTA** a Java con bucle de **Recorrido ESPECIAL**

- Con bucle for:

```
for (int i = inicio; i <= fin; avanzar(i)) {  
    if (esIgual(v[i], b)) { return true; } // Búsqueda CON éxito  
}  
// Búsqueda SIN éxito  
return false;
```

- Con bucle while:

```
int i = inicio;  
while (i <= fin) {  
    if (esIgual(v[i], b)) { return true; } // Búsqueda CON éxito  
}  
// Búsqueda SIN éxito  
return false;
```

- **esIgual** comprueba si $v[i]$ es igual a b , el dato que se busca

Búsqueda de un dato en un array


Esquema 2: traducción INDIRECTA a Java (bucle + RESOLUCIÓN BÚSQUEDA)

- Con bucle while:

```
int i = inicio; boolean esta = false;
while (i <= fin && !esIgual(v[i], b)) { avanzar(i); }
// RESOLUCIÓN DE LA BÚSQUEDA según tenga éxito o no
if (i <= fin) { esta = true; }
return esta;
```

- Con bucle for:

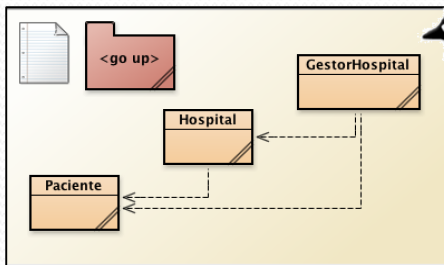
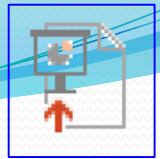
```
boolean esta = false;
for (int i = inicio; i <= fin && !esIgual(v[i], b); avanzar(i)) { ; }
// RESOLUCIÓN DE LA BÚSQUEDA según tenga éxito o no
if (i <= fin) { esta = true; }
return esta;
```



A partir de ahora, intentar resolver POR TI MISMA@ los problemas que se planteen. En cada uno de ellos...

- Sigue los pasos que se te indiquen
- Resuelve tus dudas y/o comprueba tu solución haciendo *click* en el icono del *slide-show* que existe en la transparencia

Búsqueda de un dato en un array: ejemplo



BlueJ: ejercicios - Tema 7

Completa el método **primeraLibre** de la clase **Hospital**, el único método de Búsqueda aún por diseñar en esta clase, **pues** ingresar **YA** se ha diseñado invocando a **primeraLibre**

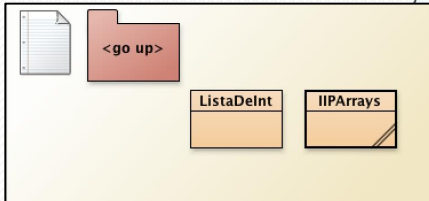
(a) **Estrategia:** si **YA** sabes que es la de Búsqueda, **ANTES** de escribir el código debes responder una pregunta... **¿Qué dato se busca?**

(b) **Traducción a Java**

```
public int primeraLibre() {
```

```
}
```


Ejercicios... ¿De Recorrido o Búsqueda? ¿Cómo estar segur@s?



BlueJ: ejercicios - Tema 7

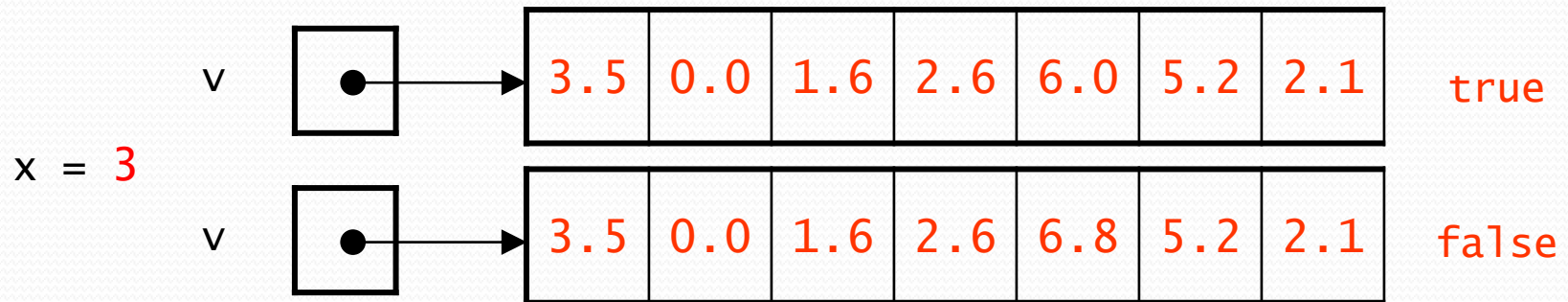
Ejercicio Nº 8 Transparencias



Diseña el método `estaElDobleDe` en la clase `IIPArrays`, que comprueba si algún elemento de un array de `double` `v` vale el doble que `x`

Preguntas Básicas, a contestar a partir del ejemplo **Y ANTES** de escribir código:

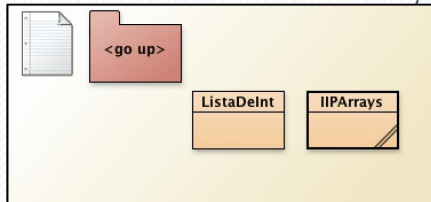
- 1.- ¿Búsqueda ó Recorrido?
- 2.- Si es Búsqueda, ¿qué dato se busca?



```
public static boolean estaElDobleDe(double x, double[] v) {
```

```
}
```

Ejercicios... ¿De Recorrido o Búsqueda? ¿Cómo estar segur@s?



BlueJ: ejercicios - Tema 7

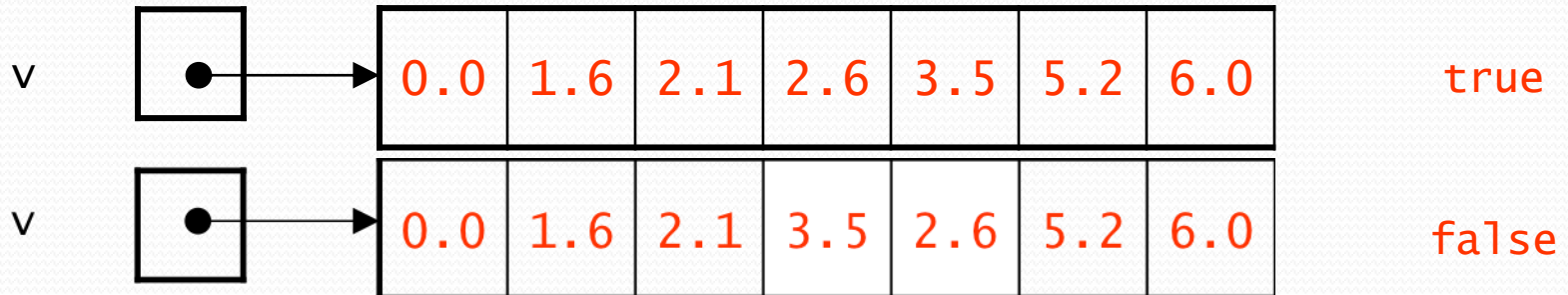
Ejercicio Nº 9 Transparencias



Diseña el método `estaEnOrdenAsc` en la clase `IIPArrays`, que comprueba si un array de `double` `v` está ordenado ascendentemente

Preguntas Básicas, a contestar a partir del ejemplo **Y ANTES** de escribir código:

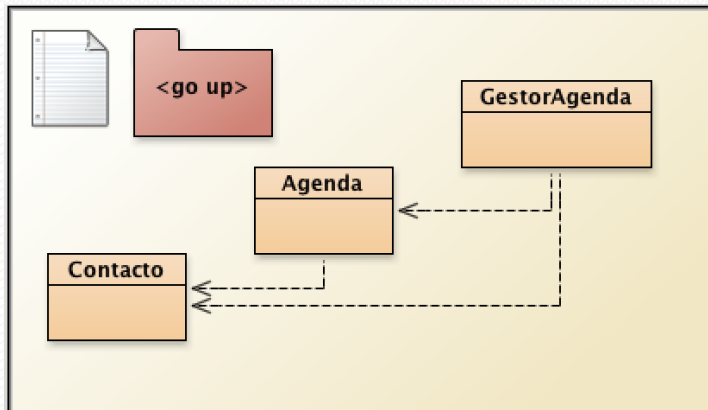
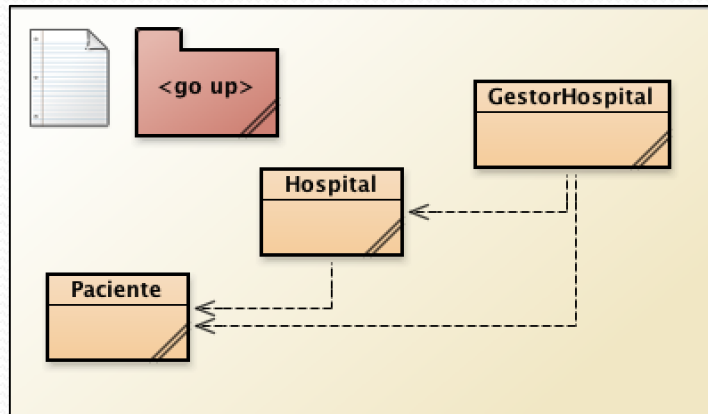
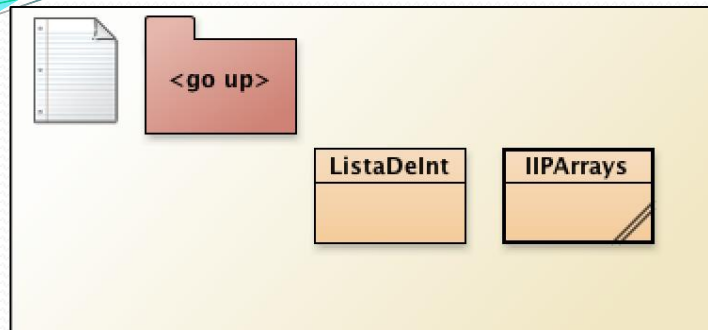
- 1.- ¿Búsqueda ó Recorrido?
- 2.- Si es Búsqueda, ¿qué dato se busca?



```
public static boolean estaOrdAsc(double[] v) {
```

```
}
```

+ Ejercicios... ¿De Recorrido o Búsqueda? ¿Combinados?



Nº 10, Nº 11, Nº 12, Nº 13 y Nº 14 Transparencias:



BlueJ: ejercicios - Tema 7

- **Completa** los métodos de Búsqueda de la clase Agenda
- **Añade** a la clase Agenda un método **posMinimo**, que devuelva la posición del primer contacto en orden lexicográfico (alfabético) por nombre de una Agenda, o **primer mínimo** de una Agenda según este orden
- **Añade** los siguientes métodos a la clase IIPArrays:
 - **capicua**, que comprueba si un array de **int v** es cap-i-cua
 - **contarBlancos**, que devuelve el nº de blancos que aparecen al final de un array de **char v**
 - **ultimoImpar**, que devuelve la posición del último impar de un array de enteros **v**, o -1 si no hay ninguno
- **Añade** los siguientes métodos a la clase ListaDeInt
 - **tresEnOrden**, que devuelve la primera posición de una lista donde se inicia una secuencia ordenada Asc. de tres Enteros, o -1 si no hay ninguna
 - **ultimoImpar**, que devuelve la posición del último impar de una lista de Enteros, o -1 si no hay ninguno
- **Añade** a la clase Hospital un método **posMaximo** que devuelva el nº de la **última** cama ocupada por un paciente sano, o **último máximo** de un Hospital según este criterio