Esta unidad es muy importante porque trata sobre las estructuras de almacenamiento, que son fundamentales a la hora de abordar la mayoría de los programas. En todos los lenguajes de programación están presentes las estructuras que se ven en la unidad, pero intenta abarcar muchas cosas, y como "quien mucho abarca…..". Por eso algunos apartados los saltaremos, bien por poco usados/útiles o por obsoletos.

Es fundamental entender el funcionamiento de estas estructuras de almacenamiento. En el material de platega muchas de ellas no están bien explicadas o no incluyen ejemplos de apoyo. Por eso, en esta unidad es muy recomendable ver en detalle estos vídeos:

Vídeo 23: Arrays I. Ejemplo muy sencillo y fácil de entender del uso de un array de una dimensión o vector. Probar a repetir vosotros el programa.

Vídeo 24: Arrays II. Introduce el bucle foreach. Repetir el programa.

Vídeos 25 y 26: Explican el uso de los arrays bidimensionales, tablas o matrices. Esta estructura para los que empezáis a programar es difícil de entender al principio. Probar el programa de los vídeos.

Vídeos 161, 162 y 163: Se ve el funcionamiento de un ArrayList. Se utilizan mucho, por lo que es importante entenderlos. Como anteriormente, escribir vosotros el código de estos vídeos.

Vídeo 186. Uso de TreeSet.
Vídeo 189. Uso de Maps.

Adjunto también un pdf con ejercicios hechos en el módulo presencial que os vendrá bien como material complementario. Ese pdf contiene las colecciones mas utilizadas. No quiere decir, ni mucho menos que el resto no sean importantes, pero en un primer curso de programación es imposible verlas todas en profundidad.

También en el libro "Java como programar de Deytel" (yo tengo la versión 9 del libro) viene muy bien explicada la teoría en los siguientes capítulos:

Capítulo 7: Arreglos y objetos ArrayList.

Capítulo 20: Colecciones genéricas.

Con todo esto y el boletín de ejercicios resueltos, pienso que lo vais a tener mas fácil para entender la unidad.

Ordenación de ArrayList de objetos:

Ejemplo de cómo ordenar un ArrayList de objetos de la clase Persona por el valor de la edad. En los enlaces siguientes hay información complementaria:

http://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=587:ejercicio-ejemplo-resuelto-interface-comparable-y-metodo-compareto-api-java-comparar-objetos-cu00911c&catid=58:curso-lenguaje-programacion-java-nivel-avanzado-i&Itemid=180

https://jarroba.com/ordenar-un-arraylist-en-java/

```
class Persona implements Comparable<Persona> {

   private String nombre;
   private int edad;

   public Persona() {
   }

   public Persona(String nombre, int edad) {
      this.nombre = nombre;
      this.edad = edad;
   }

   public String getNombre() {
      return nombre;
   }

   public void setNombre(String nombre) {
      this.nombre = nombre;
   }

   public int getEdad() {
      return edad;
   }

   public void setEdad(int edad) {
      this.edad = edad;
   }

   @Override
   public String toString() {
      return this.getNombre() + "  -  " + this.getEdad();
   }

   public int compareTo(Persona p) {
      int resultado = 0;

      if (this.edad < p.edad) {
```

```
            resultado = -1;
        } else if (this.edad > p.edad) {
            resultado = 1;
        } else {
            resultado = 0;
        }
        return resultado;
    }
}

import java.util.ArrayList;
import java.util.Collections;


public class Ordenar {
    public static void main(String[] args) {
        ArrayList listaPersonas = new ArrayList<>();

        listaPersonas.add(new Persona("Marisa",26));
        listaPersonas.add(new Persona("Juan",16));
        listaPersonas.add(new Persona("Lola",18));

        Collections.sort(listaPersonas);

        for (Object listaPersona : listaPersonas) {
            System.out.println(listaPersona.toString());
        }
    }
}
```

No ver en el material de platega los siguientes apartados:

2.1.4.- Operaciones avanzadas con cadenas de caracteres (V).

2.2, 2.2.1 y 2.2.2.- Expresiones regulares.

5 Clases y métodos genéricos. Las colecciones los utilizan, pero lo que viene en este apartado no es necesario.

10 iteradores

12 Tratamiento de documentos estructurados XML.