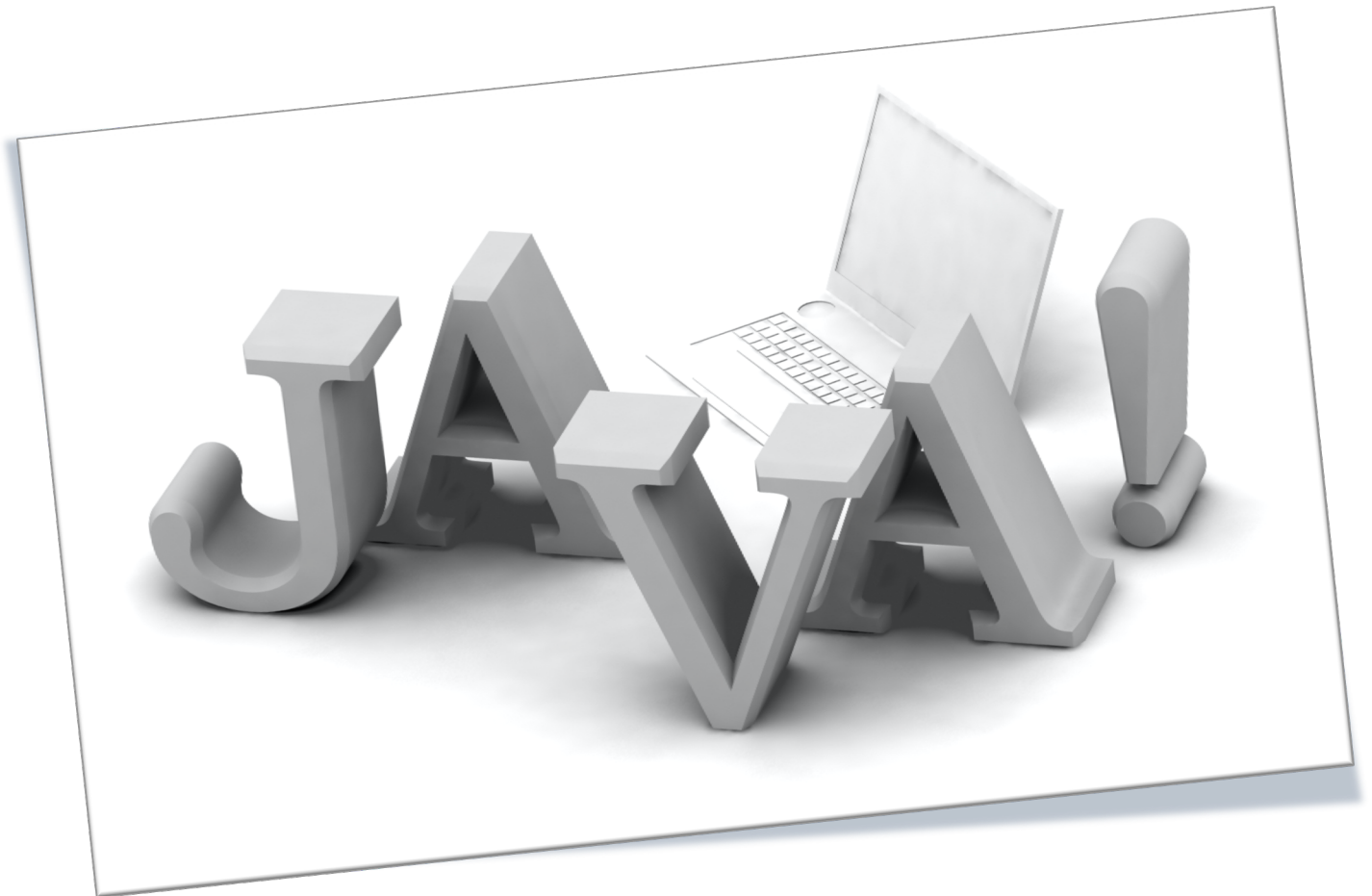

PRÁCTICAS

Capítulo 6: Programación gráfica y ficheros



Unidad 6. Práctica 6.3 Uso de librerías externas: html, json, doc, excel, pdf

Crea un proyecto que se llame Capítulo6

El proyecto debe estar en el paquete org.iesjaroso.daw.capitulo6.

Organiza los ejercicios en paquetes si lo consideras necesario.

Debemos considerar en realizar las prácticas en diferentes proyectos

JSoup. Documento HTML, XML, JSON

Sitio de descarga: <https://jsoup.org/>

Manual y tutorial: <https://jsoup.org/cookbook/>

Tutorial: <https://www.adictosaltrabajo.com/2019/05/14/implementando-un-crawler-sencillo-con-jsoup/>

Ejercicio 1.- Tratamiento de documentos HTML y extracción de web. Crea un proyecto llamado Capitulo6Jsoup que capture las siguientes webs y realiza la actividad asignada utilizando la biblioteca Jsoup

- En la web <https://www.nombra.me> contiene los nombres más usuales y su significado. Se pide extraer dichos nombres junto con su significado. Para ello se deberá recorrer todos los enlaces.

Fíjate que la estructura de enlaces es la siguiente:

<https://www.nombra.me/nombres/espa%C3%B1ol.html?page=45>

Dónde page son las páginas que existen. Almacénalos en formato html usando Jsoup

- Extrae al menos 100 recetas de cocina de la web <https://www.recetasgratis.net/busqueda/q/pasta/cat/5> Cada receta debe incluir el título, ingredientes y pasos a seguir. Almacénalos en formato Word usando las librerías vistas en clases. Deben incluir la imagen.

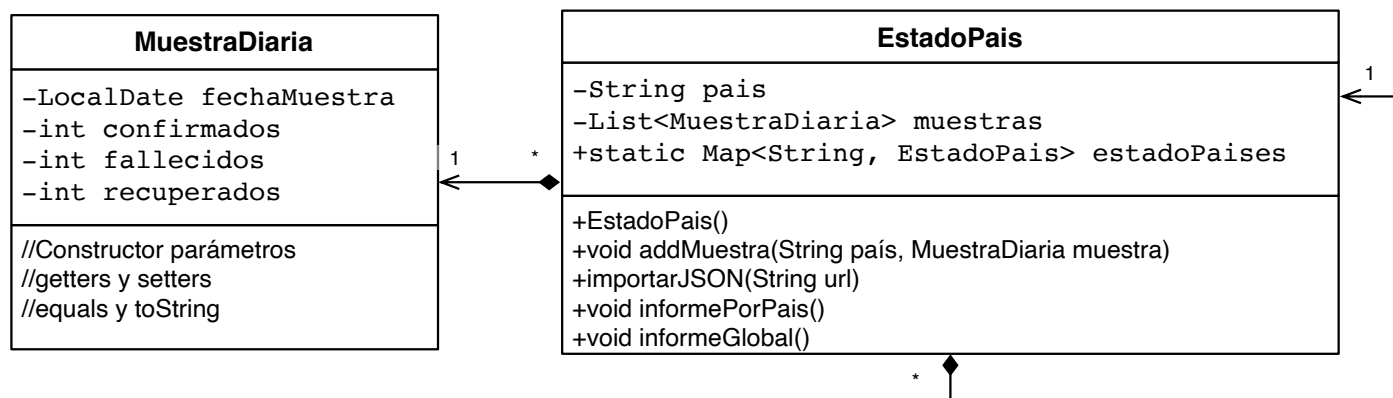
Documento JSON. Biblioteca gson.

Junto a la práctica tienes un tutorial de gson y la última versión de la librería gson.

Sitio de descarga: <https://github.com/google/gson/blob/master/UserGuide.md>

Tutorial: <https://www.adictosaltrabajo.com/2012/09/17/gson-java-json/>

Ejercicio 2.- Tratamiento de documentos JSON. Crea un proyecto llamado Capitulo6Gson. Se desea crear una aplicación que permita hacer un seguimiento diario de la evolución del coronavirus. Para ello se han diseñado las siguientes clases:



La clase MuestraDiaria tiene la información de cada día con los datos:

- ✓ Fecha del día cuando se toma la muestra
- ✓ Los casos confirmados
- ✓ El número de fallecidos
- ✓ El número de recuperados

Crea los métodos:

- ✓ Constructor con todos los parámetros
- ✓ Getters
- ✓ Setters con validaciones
- ✓ Sobrescribe toString. Mostrará todos los datos
- ✓ Sobrescribe equals. Dos muestras son iguales si todos los datos son iguales
- ✓ Establece orden natural por fechaMuestra

En la clase EstadoPais tenemos el nombre los datos:

- ✓ Nombre del país
- ✓ Todas las muestras hasta la fecha.
- ✓ Un mapa static donde se almacenan todos los datos de los países. El String del mapa es el nombre del país y EstadoPais es un objeto de EstadoPais con la información de las muestras de un determinado país.

Crea los métodos:

- ✓ Constructor() Debe inicializar las estructuras muestras y estadoPaises. También es recomendable añadir todos los países al mapa estadoPaises con la lista vacía o hacerlo desde importarJSON
- ✓ +void addMuestra(String pais, MuestraDiaria muestra): añade una muestra a un país
- ✓ +importarJSON(String url). Importa los datos desde una URL en formato JSON. Puedes cambiar esta función por importar de un fichero local en formato .json
Para importar los datos de una URL
- ✓ +void informePorPais(): Muestra la información actual de un país. Para ello debe tomar todas las muestras y hacer el calculo total de confirmados, fallecidos y recuperados (curados).
Observa que una muestra solo tiene información de un determinado día.
- ✓ +void informeGlobal(): Muestra la información de todos los países.

La información está disponible en la ubicación: <https://pomber.github.io/covid19/timeseries.json> en formato json con la siguiente estructura:

```
{
  "Thailand": [
    {
      "date": "2020-1-22",
      "confirmed": 2,
      "deaths": 0,
      "recovered": 0
    },
    {
      "date": "2020-1-23",
      "confirmed": 3,
      "deaths": 0,
      "recovered": 0
    },
    ...
  ],
  ...
}
```

Ejemplo de uso de la librería GSON

```
static class Item {
    String title;
    String link;
    String description;
}

static class Page {
    String title;
    String link;
    String description;
    String language;
    List<Item> items;
}

public static void main(String[] args) throws Exception {

    String json = readUrl("http://www.javascriptkit.com/"
        + "dhtmltutors/javascriptkit.json");

    Gson gson = new Gson();
    Page page = gson.fromJson(json, Page.class);

    System.out.println(page.title);
    for (Item item : page.items)
        System.out.println("    " + item.title);
}
```

Realiza un método que permita responder a las preguntas o hazlo directamente desde el main:

- ✓ 10 primeros países dónde se originó el contagio.
- ✓ 3 últimos países en contagiarse.
- ✓ ¿Qué día hubo más contagios?
- ✓ ¿Qué 5 países van frenando el número de contagios?
- ✓ Muestra por orden cronológico (por fecha) los países que fueron contagiándose?
- ✓ ¿En qué 5 países es más alto el nivel de mortalidad según los contagios?
- ✓ Mostrar los países con porcentaje de curados respecto a los contagiados

Documentos PDF.

Sitio de descarga: Son necesarias cualquiera de las librerías siguientes:

- ✓ **iText**: <https://search.maven.org/classic/#search%7Cga%7C1%7Ca%3A%22itextpdf%22>
- ✓ **PdfBox**: <https://search.maven.org/classic/#search%7Cga%7C1%7Ca%3A%22pdfbox%22>

Consideraciones: Ten en cuenta que son 2 bibliotecas diferentes.

Tutorial:

- ✓ <https://www.baeldung.com/java-pdf-creation>
- ✓ <http://codigoxules.org/java-itext-pdf-creando-pdf-java-itext/>

Ejercicio 3.- Tratamiento de documentos PDF. Crea un proyecto llamado Capitulo6PDFDoc que partiendo del ejercicio2, realice las tareas:

- a) Almacene datos en un fichero pdf. Inserta títulos, cambia la tipografía e inserta una imagen.
- b) Abrir un documento en pdf y mostrar el contenido por pantalla. (Solo el texto)

NOTA.

- ✓ Para ello necesitarás utilizar las librerías de PDF iText y PdfBox.
- ✓ Cuidado al realizar las importaciones. Asegúrate que son de com.itextpdf.text.*

Documentos word(.docx) y excel (.xlsx).

Ejercicio 4.- Tratamiento de documentos Word y Excel. Crea un proyecto llamado Capitulo6Office y realiza las tareas:

- ✓ Abrir un fichero .docx
- ✓ Guardar un fichero .docx
- ✓ Abrir un fichero Excel .xlsx
- ✓ Guardar un fichero Excel