Servlets -Ejercicios de iniciación

Proyecto Java Web con ejercicios de iniciación a la programación de servlets
Su página principal es el archivo **index.html** o index.jsp con un enlace cada uno de los ejercicios:
Ejercicio 1 Ejercicio 2 Ejercicio 3 ….

Ejercicio 1

El formulario de **conversorCF.html** se envía por GET a un servlet **ServletConversor** que visualiza:

- El resultado de la conversión, o un mensaje de error, según corresponda
- Un enlace para volver al formulario

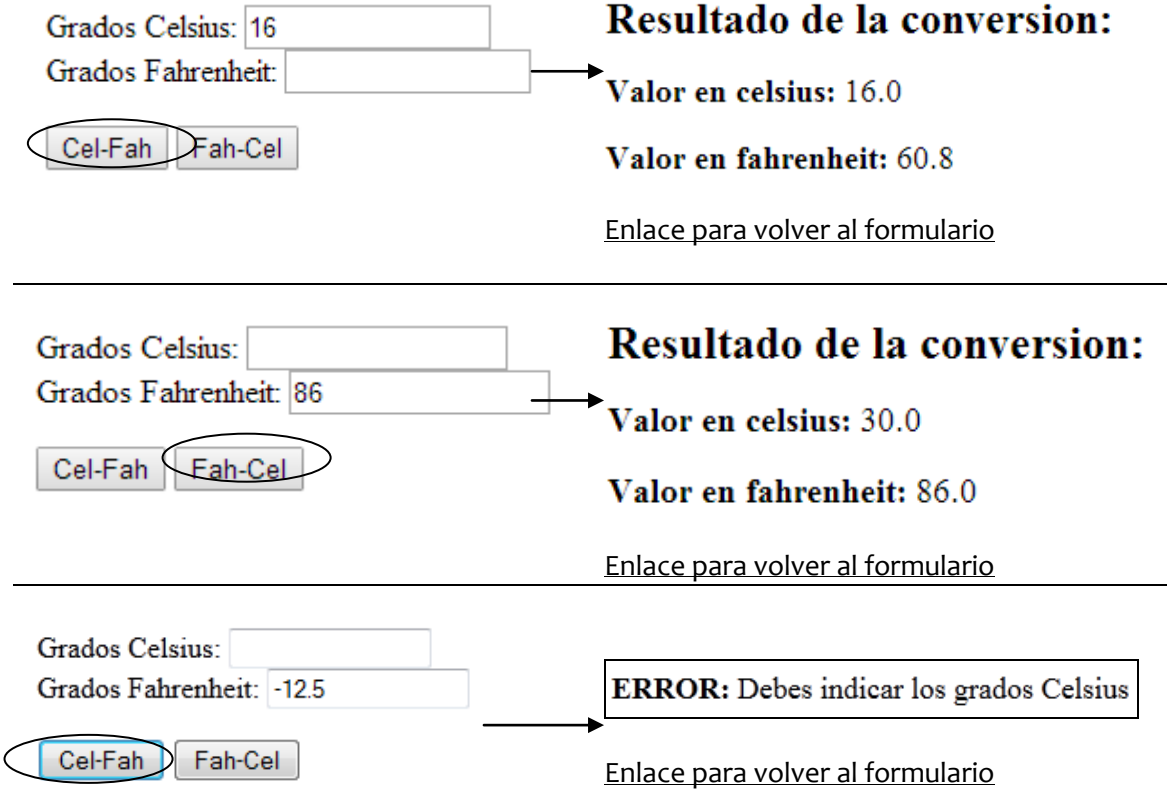
ServletConversor no es accesible por el nombre de la clase, sino por cualquiera de estos patrones:

- "convertirtemperatura"
- cualquier ruta que comience por "conversion/"

$$°C \times 9/5 + 32 = °F$$

$$(°F - 32) \times 5/9 = °C$$

conversorCF.html **ServletConversor**



Ejercicio 1: Cambio 1

Modifica el ejercicio anterior para que **un bean** se encargue de la conversión.
(Ahora el servlet, además de ocuparse de la salida HTML, es puente entre el cliente y la lógica de negocio (el bean))
Crea una clase Java llamada **ConversionCF**, y úsala desde el servlet para la conversión.
ConversionCF tiene:

- 2 atributos: celsius y fahrenheit
- Un constructor que recibe una temperatura y su tipo ('c' para Celsius, 'f' para Fahrenheit), y carga los 2 atributos con los valores adecuados
- Métodos para recoger el valor de los atributos (getters)

Ejercicio 1: Cambio 2

Ahora queremos que el servlet **ServletConversor**, realice una tarea adicional: guardar en una colección los distintos "locale"s (o configuraciones de idioma) de los clientes que se le van conectando.

- Dicha colección será un **HashSet** de Strings.

  Si, en un momento dado de la vida del Servlet se han conectado 100 personas de Reino Unido, 200 de España y 3 de Francia, el HashSet contendrá:

UK	ES	FR

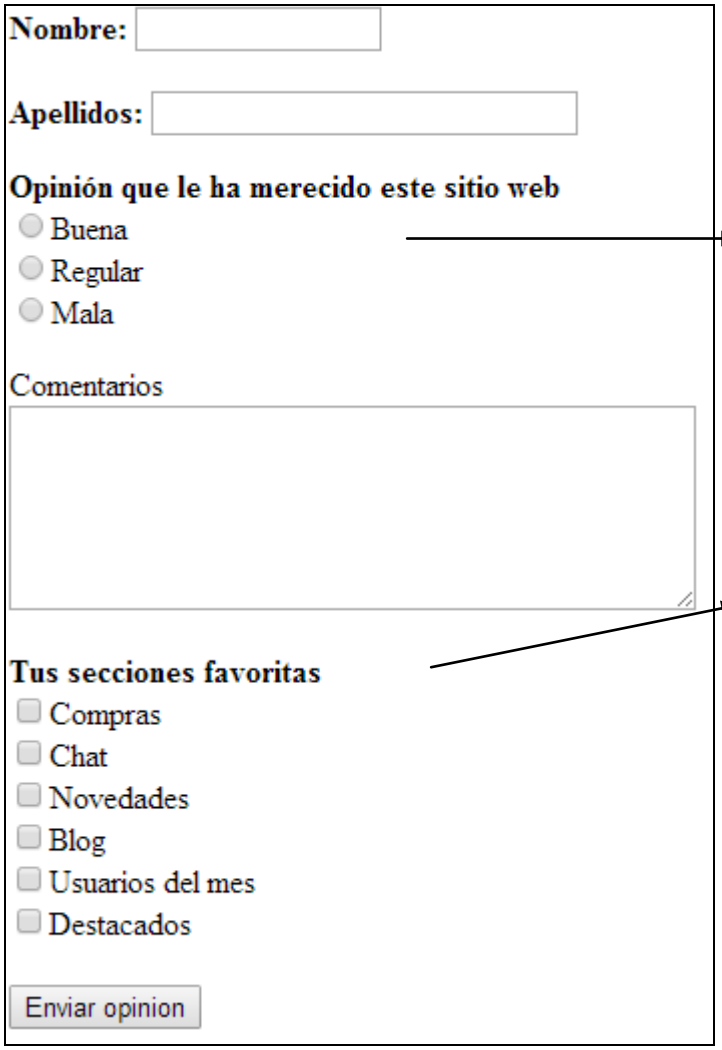
  Un HashSet es una colección Java similar a un ArrayList (se instancia igual, métodos add, remove,….), pero:
    - Es de tipo conjunto (Set): no permite elementos duplicados y los elementos no se guardan ordenados
    - las búsquedas en él son más rápidas

- El Servlet mostrará, en su parte inferior:
  "Se han establecido conexiones desde 3 distintos locale's"

(Para comprobar que funciona, puedes cambiar temporalmente el idioma de algún navegador – sin parar el servlet-)

Ejercicio 2

Partirá del servlet **ServletFormOpinion,** encargado de dibujar el siguiente formulario, que se envía a sí mismo por POST



Grupo de radios con valores B, R, M
En un principio ninguno está seleccionado

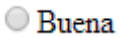
Estas secciones son variables. Se extraen de un fichero de texto **secciones.txt** (con una sección por línea), que deberás crear a mano en la sección

Las clases Java de un proyecto JavaEE no tienen acceso directo a los recursos estáticos, por lo que para acceder al fichero de texto "secciones" desde el Servlet deberás usar esta ruta:

...getServletContext().getRealPath("fich.txt");

- Si al enviar la opinión, se da algún error (nombre vacío o ningún radio seleccionado), se volverá a mostrar de nuevo el formulario, además de un mensaje de error en la parte superior

- Si no hay errores:

  -Y además la opinión es 

, se grabará una nueva línea en un fichero de texto **seccionesfavoritas.txt** con el nombre del usuario y una concatenación de sus secciones favoritas, tipo: "Juan Lopez: Compras,Blog, Destacados".

  (Para verificar el contenido del fichero, debes ver el proyecto desplegado en: 

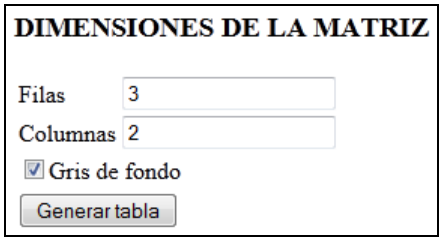
  -Se volverá a cargar el formulario vacío

Ejercicio 4

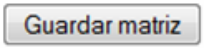
Página **index.html**, con el siguiente formulario que será enviado al Servlet **IntroCeldas**

IntroCeldas mostrará un formulario que representa una matriz matemática en la que el usuario introducirá datos. Dicha matriz tendrá el Nº de filas y columnas solicitadas en index.html, albergando cada celda un inputbox de nombre (name) celda1-1, celda1-2, etc
IntroCeldas se ayudará de un método dibujaMatriz que se encarga de dibujar la tabla con los input box de la matriz en la respuesta.
Si el usuario deja vacía alguna de las cajas de texto Filas o Columnas de index.html, IntroCeldas redirigirá la respuesta a index.html, (mediante el método sendRedirect de la misma).

index.html



Servlet **IntroCeldas**

El usuario rellenará el formulario/matriz de **IntroCeldas** con los valores deseados y al pinchar 

se enviará a otro servlet denominado **GuardaMatriz.**

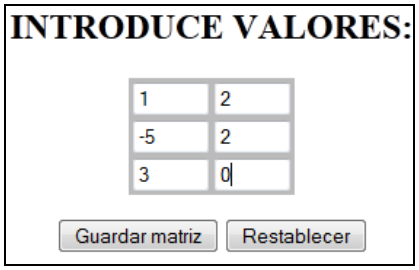
Este 2º servlet (**GuardaMatriz**) hará lo siguiente:

- En caso de haberse dejado alguna celda de la matriz vacía, o con algún valor no numérico, mostrará el mensaje "Debes rellenar correctamente la matriz"
  Debes utilizar excepciones para detectar esta situación
- En caso de haberse rellenado correctamente las cajas de texto, generará una matriz (int[][]) y la añadirá a un conjunto de matrices haciendo uso de la clase **AlmacenMatrices.java**:

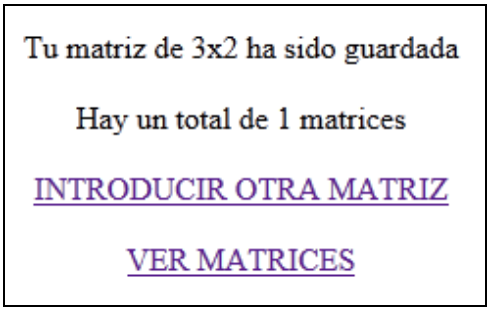
  ***** Para guardar todas las matrices se crea una nueva clase **AlmacenMatrices** con un atributo estático que es un ArrayList de matrices bidimensionales de int. Esta clase deberá incluir los métodos que estimes necesarios.

  Así pues, el servlet GuardaMatriz:
  - Elaborará una matriz con los datos del formulario y la añadirá al ArrayList estático de la clase AlmacenMatrices
  - Mostrará el mensaje "Tu matriz de NxM ha sido guardada"
  - Mostrará cuántas matrices hay guardadas en este momento
  - Mostrará 2 enlaces: 1 a index.html para proceder a introducir otra matriz, otro a un nuevo Servlet **VisorMatrices**

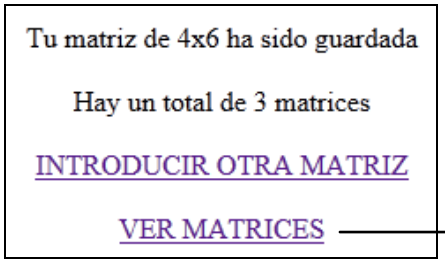
Servlet **IntroCeldas**



Servlet **GuardaMatriz**



El Servlet **VisorMatrices,** visualizará las matrices que ha ido añadiendo el usuario y que están almacenadas en el ArrayList. Para visualizar cada una de ellas en forma de tabla llamará a un método



Servlet **VisorMatrices**