Ejercicio 1.

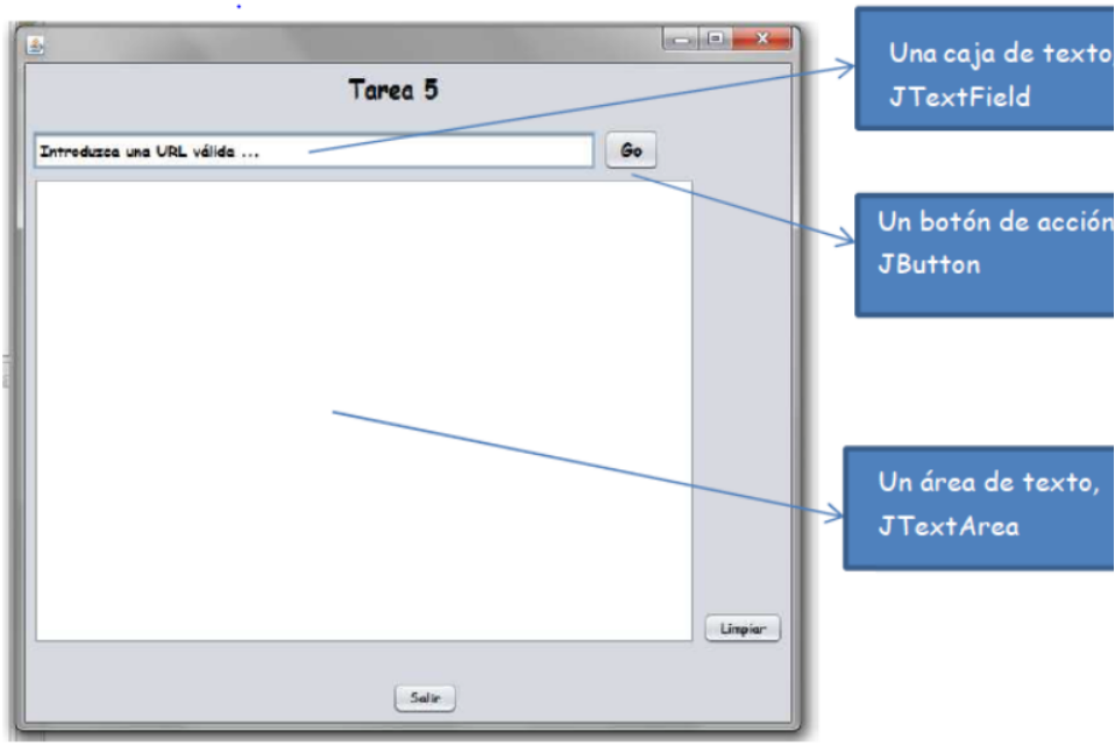
Modifica el ejemplo del servidor HTTP (Proyecto java `ServerHTTP`, apartado 5.1 de los contenidos) para que implemente multihilo, y pueda gestionar la concurrencia de manera eficiente.

Ejercicio 2

El objetivo es diseñar una aplicación Java que utilice URLConnection, simulando el comportamiento de un navegador web básico.

La aplicación tendrá una interfaz gráfica Swing sencilla a partir de un formulario JFrame.

El formulario incluirá una caja de texto (JTextField), un área de texto (JTextArea) y tres botones de accion (JButton). La apariencia será similar a esta:



Descripción del funcionamiento:

- El programa debe permitir introducir una URL en la caja de texto (JTextField).
- Al presionar el botón "Go", la aplicación intentará conectar con la URL escrita en la caja de texto. En función del tipo MIME pdf o html descargará una página html o un fichero pdf. Más concretamente se hará lo siguiente:
    -

    - Si la URL no es válida, mostrará un mensaje con el error producido en el área de texto (JTextArea). El mensaje se verá a continuación de los mensajes que ya haya escritos.
    - Si la URL es válida, se analizará el contenido de la misma y:
        - Si la URL es del tipo "application/pdf",será necesario programar la descarga del contenido del archivo pdf. Puede servirte de ayuda el ejemplo de "Accesos a Recursos URL y URL Connection" del apartado 3.6 del tema.
        - Si la URL es del tipo "text/html", se mostrará en el área de texto (JTextArea) el contenido del documento representado en la URL. No será necesario mostrar el contenido de la URL con ningún formato (sería tan complejo como desarrollar un navegador); se trata de inspeccionar el texto del contenido de la URL y mostrarlo línea a línea a continuación de los mensajes que ya haya en el área de texto.
    - En ambos casos la aplicación mostrará en el área de texto (JTextArea) de la aplicación la siguiente información de la cabecera de la URL:
        - Tipo de contenido.
        - Longitud.
        - Fecha de la última modificación.
    - El comportamiento del botón Go se hace en un nuevo hilo. En este caso habrá que crear una nueva clase "HiloBoton" que herede de Thread. En su constructor recibirá la URL recogida del JTextField y el objeto JTextArea para que pueda escribir en el.
- El botón "Limpiar" se encargará de limpiar y vaciar el contenido del área y de la caja de texto (JTextArea y JTextField) del formulario.
- El botón "Salir" terminará la aplicación.

Criterios de puntuación. Total 10 puntos.

- Ejercicio 1 - 3 puntos.
    - Codificación correcta cumpliendo con los requisitos del enunciado (1 punto).
    - Funcionamiento según las especificaciones (1 punto).
    - Documentación del cambio realizado (1 punto).

- Ejercicio 2 - 7 puntos.
    - Codificación correcta cumpliendo con los requisitos del enunciado (4 puntos).
    - Funcionamiento según las especificaciones (1 puntos).
    - Implementación del multihilo (0,5 puntos)
    - Tratamiento adecuado de las posibles excepciones (1 puntos).
    - Documentación (0,5 puntos).

Indicaciones para la entrega

Una vez realizada la tarea tienes que subir una carpeta comprimida con los proyectos y las aplicaciones. El envío se realizará a través de la plataforma de la forma establecida para ello, y el archivo se nombrará siguiendo las siguientes pautas:

apellido1_apellido2_nombre_SIGxx_Tarea

Asegúrate que el nombre no contenga la letra ñ, tildes ni caracteres especiales extraños.
Así por ejemplo la alumna Begoña Sánchez Mañas para la segunda unidad
del MP de PSP, debería nombrar esta tarea como...

sanchez_manas_begona_PSP02_Tarea