Tarea Individual 23 - Conexión FTP

El objetivo de esta actividad es desarrollar un programa que permita la conexión a un servidor FTP, la autenticación mediante un usuario y contraseña, la obtención de información sobre los archivos y directorios disponibles en el servidor, y la implementación de un cierre de sesión adecuado.

Descripción de la Tarea

Con base en el código proporcionado, se solicita a los alumnos realizar las siguientes modificaciones y desarrollos:

- **Conexión al servidor FTP:**
  - Utilizar la clase `FTPClient` de la biblioteca `org.apache.commons.net.ftp` para establecer la conexión al servidor FTP especificado.
  - Configurar el cliente en modo pasivo mediante `enterLocalPassiveMode`.

```java
import java.io.IOException;
import org.apache.commons.net.ftp.FTPClient;
import org.apache.commons.net.ftp.FTPFile;

public class FTPConexion {


    private static final String SERVER = "ftp.rediris.es"; // Reemplazar con el servidor FTP real
    private static final int PORT = 21; // Puerto predeterminado para FTP
    private static final String USER = "usuario";
    private static final String PASSWORD = "contraseña";

    public static void main(String[] args) {
        FTPClient ftpClient = new FTPClient();

        try {
            // Conexión al servidor FTP
            System.out.println("Intentando conectar al servidor FTP...");
            ftpClient.connect(SERVER, PORT);
            ftpClient.enterLocalPassiveMode(); // Configuración en modo pasivo
```

- **Autenticación:**
  - Implementar el inicio de sesión con las credenciales proporcionadas (`usuario` y `clave`).
  - Mostrar mensajes en la consola indicando si el inicio de sesión fue exitoso o fallido.

```java
            // Autenticación
            if (ftpClient.login(USER, PASSWORD)) {
                System.out.println("Inicio de sesión exitoso.");
            } else {
                System.out.println("Inicio de sesión fallido. Verifique las credenciales.");
                return; // Salir si la autenticación falla
            }
```

- **Listar archivos y directorios:**
  - Utilizar el método `listFiles` para obtener un arreglo de objetos `FTPFile` que representen los elementos en el directorio actual.
  - Asignar una descripción de tipo (`Fichero`, `Directorio`, `Enlace simbólico`) a cada elemento y mostrar su nombre y tipo en la consola.

```
            // Listar archivos y directorios
            System.out.println("Obteniendo lista de archivos y directorios...");
            FTPFile[] files = ftpClient.listFiles();
            if (files.length == 0) {
                System.out.println("El directorio está vacío.");
            } else {
                for (FTPFile file : files) {
                    String fileType = getTypeDescription(file);
                    System.out.println("Nombre: " + file.getName() + ", Tipo: " + fileType);
                }
            }
```

- **Cierre de sesión y desconexión:**
    - Implementar el cierre de sesión con `logout` e indicar en la consola si fue exitoso o no.
    - Asegurarse de cerrar correctamente la conexión con el servidor utilizando `disconnect`.

```
// Cierre de sesión y desconexión
if (ftpClient.logout()) {
    System.out.println("Cierre de sesión exitoso.");
} else {
    System.out.println("No se pudo cerrar la sesión.");
}

ftpClient.disconnect();
System.out.println("Conexión cerrada.");
```

- **Requisitos adicionales:**
    - Gestionar las posibles excepciones (`SocketException`, `IOException`) mostrando mensajes claros en la consola.
    - Comentar el código para explicar las funcionalidades principales.
    - Utilizar un array para asociar los códigos de tipo de archivo (`Fichero`, `Directorio`, `Enlace simbólico`) a los elementos listados.

```
        } catch (IOException e) {
            System.err.println("Error de conexión o comunicación con el servidor FTP: " + e.getMessage());
        } finally {
            try {
                if (ftpClient.isConnected()) {
                    ftpClient.disconnect();
                    System.out.println("Desconexión forzada debido a un error.");
                }
            } catch (IOException e) {
                System.err.println("Error al desconectar del servidor FTP: " + e.getMessage());
            }
        }
    }

    /**
     * Obtiene una descripción del tipo de archivo basada en su código.
     *
     * @param file Objeto FTPFile que representa un archivo o directorio.
     * @return Descripción del tipo de archivo (Fichero, Directorio, Enlace simbólico).
     */
    private static String getTypeDescription(FTPFile file) {
        if (file.isDirectory()) {
            return "Directorio";
        } else if (file.isFile()) {
            return "Fichero";
        } else if (file.isSymbolicLink()) {
            return "Enlace simbólico";
        } else {
            return "Tipo desconocido";
        }
    }
```