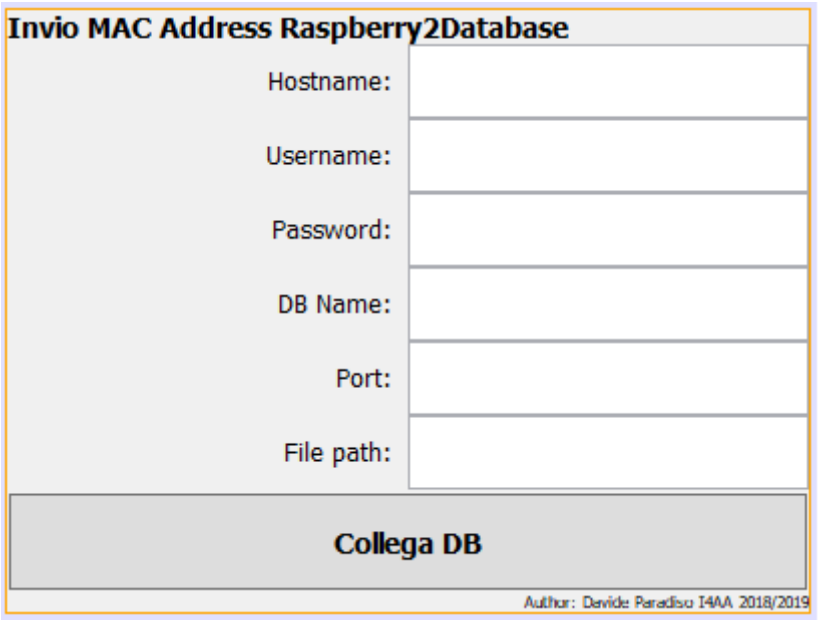
Diario di lavoro

Luogo	Scuola Arti e Mestieri Trevano
Data	mercoledì, 17 ottobre 2018

Lavori svolti

Oggi ho scritto il codice dell'applicativo per collegarsi al database e inviare i mac address tramite un file di testo.



Se l'utente clicca su **Collega DB** avviene la connessione al Database e l'invio dei dati:

```
    private void collegaDB(java.awt.event.ActionEvent evt) {
        String hostname = jtHostname.getText();
        String username = jtUsername.getText();
        String password = new String(jPasswordFieldl.getPassword());
        String dbName = jtDBName.getText();
        String path = jtFile.getText();
        String port = jtPort.getText();
        DBConnection.invioDati(hostname, username, password, dbName, port, path, this);
    }
```

Per fare ciò ho creato la classe statica **DBConnection** con i suoi metodi principali:

```java
/**
 * Metodo che legge il file e ritorna un ArrayList con il suo contenuto.
 * Se il file é vuoto ritorna null.
 *
 * @param file File da leggere
 * @return ArrayList con il contenuto del file
 * @throws FileNotFoundException
 * @throws IOException
 */
private static ArrayList<String> leggiFile(File file) {
    BufferedReader reader = null;
    ArrayList<String> lines = null;
    try {
        // Oggetto per leggere il file
        reader = new BufferedReader(new FileReader(file));
        // Array dove memorizzo i MAC Address
        lines = new ArrayList<>();
        String currentLine;
        // Prendo tutti i MAC Address
        while ((currentLine = reader.readLine()) != null) {
            lines.add(currentLine);
        }
        reader.close();
    } catch (FileNotFoundException ex) {
        return null;
    } catch (IOException ex) {
        return null;
    }
    if (lines.isEmpty() || lines == null) {
        return null;
    }
    return lines;
}
/**
 * Metodo che svuota il contenuto del file
 *
 * @param file File da svuotare
 */
private static void svuotaFile(File file) {
    PrintWriter writer = null;
    try {
        writer = new PrintWriter(file);
        writer.print("");
        writer.close();
    } catch (FileNotFoundException ex) {
        Logger.getLogger(DBConnection.class.getName()).log(Level.SEVERE, null, ex);
    }
}
```

Questo purtroppo ancora non funziona:

```
/**
 * Metodo che fa la connessione al database e la ritorna, se non riesce a
 * farla ritorna null.
 *
 * @param hostname Nome del server
 * @param username Username del database
 * @param password Password dello username
 * @param dbName Nome del database
 * @param port Porta con cui connettersi al database
 * @return Ritorna la connessione al database
 * @throws ClassNotFoundException
 * @throws SQLException
 */
private static Connection connessioneDB(String hostname, String username, String password, String dbName, int port) {
    Connection con = null;
    try {
        Class.forName("oracle.jdbc.driver.OracleDriver").newInstance();
        // Connessione al Database
        String c = "jdbc:mysql://" + hostname + ":" + port + "/" + dbName;
        con = DriverManager.getConnection(
                c,
                username,
                password
        );

    } catch (SQLException | ClassNotFoundException | InstantiationException | IllegalAccessException ex) {
        System.out.println(ex);
        return null;
    }
    return con;
}
```

```java
public final class DBConnection {

    /**
     * Metodo che invia il contenuto del file con i MAC Address al database
     *
     * @param hostname Nome del server
     * @param username Username del database
     * @param password Password dello username
     * @param dbName Nome del database
     * @param port Porta con cui connettersi al database
     * @param pathFile Path del file dove andare a prendere i dati
     * @param frame frame che richiama il metodo
     * @return Se il processo é stato fatto correttamente ritorna true
     */
    public static boolean invioDati(
            String hostname, String username, String password,
            String dbName, String port, String pathFile, JFrame frame
    ) {
        int iPort = 3306;
        try {
            iPort = Integer.parseInt(port);
        } catch (NumberFormatException e) {
            JOptionPane.showMessageDialog(frame, "Inserire una porta che sia un numero intero, di default é la 3306");
        }
        try {
            // Connessione al Database, se non mi conetto esco
            Connection con = connessioneDB(hostname, username, password, dbName, iPort);
            if (con == null) {
                JOptionPane.showMessageDialog(frame, "Credenzali per la connessione al Database errate");
                return false;
            }
            // Oggetto per fare delle query
            Statement stmt = con.createStatement();
            // File da cui prendere i dati
            File file = new File(pathFile);
            ArrayList<String> lines = null;
            // Leggo il contenuto del file, se il file é vuoto esco
            lines = leggiFile(file);
            if (lines == null) {
                JOptionPane.showMessageDialog(frame, "File vuoto");
                return false;
            }
            // Query per inviare i dati al DB
            for (int i = 0; i < lines.size(); i++) {
                ResultSet result = stmt.executeQuery(
                        "INSERT INTO address(MAC_Address,BlackList,deleted) VALUES(" + lines.get(i) + ",0,0)"
                );
            }
            // Dopo aver fatto l'insert svuoto il file
            svuotaFile(file);
            return true;
        } catch (SQLException ex) {
            JOptionPane.showMessageDialog(frame, "Non é possibile fare la query");
            return false;
        }
    }
```

Come ultima cosa ho creato il metodo per fare il loop del processo d'invio ogni 10 minuti:

```java
private static void loopSend(ArrayList<String> lines, File file, JFrame frame, String dbName, Statement stmt ) {
    try {
        // Leggo il contenuto del file, se il file é vuoto esco
        lines = leggiFile(file);
        if (lines == null) {
            JOptionPane.showMessageDialog(frame, "File vuoto");
        }
        // Query per inviare i dati al DB
        for (int i = 0; i < lines.size(); i++) {
            String query = "INSERT INTO " + dbName + ".address(MAC_Address,BlackList,deleted) VALUES('" + lines.get(i) + "',0,0)";
            stmt.executeUpdate(query);
        }
        // Dopo aver fatto l'insert svuoto il file
        svuotaFile(file);
        Thread.sleep(6000000);
    } catch (InterruptedException | SQLException ex) {

    }
}
```

Problemi riscontrati e soluzioni adottate

Il metodo **connessioneDB** da un errore, non trova una libreria che ho scaricato e importato del progetto. Devo ancora capire come risolverlo.
Nel pomeriggio ho risolto il problema, ho scaricato dal sito ufficiale di oracle la libreria che mi serviva e ora il codice del metodo **connessioneDB** é così:

```
/**
 * Metodo che fa la connessione al database e la ritorna, se non riesce a farla ritorna null.
 *
 * @param hostname Nome del server
 * @param username Username del database
 * @param password Password dello username
 * @param dbName Nome del database
 * @param port Porta con cui connettersi al database
 * @return Ritorna la connessione al database
 * @throws ClassNotFoundException
 * @throws SQLException
 */
private static Connection connessioneDB(String hostname, String username, String password, String dbName, int port) {
    Connection con = null;
    try {
        String c = "jdbc:mysql://" + hostname + ":" + port + "/" + dbName +
                "?useUnicode=true&useJDBCCompliantTimezoneShift=true&useLegacyDatetimeCode=false&serverTimezone=UTC";
        con = DriverManager.getConnection(
                c,
                username,
                password
        );

    } catch (SQLException ex) {
        System.out.println(ex);
        return null;
    }
    return con;
}
```

Punto della situazione rispetto alla pianificazione

Secondo la mia preventiva pianificazione leggermente in ritardo.

Programma di massima per la prossima giornata di lavoro

Testare l'applicativo più a fondo