

```

#include <iostream>
#include <fstream>
# include <list>

#include "iPud.h"

using namespace std;

bool resuelveCaso() {

    string comando, c, a;
    int d, n;

    cin >> comando;

    if (!cin) {
        return false;
    }
    else {

        sistema ipud;

        while (comando != "FIN") {
            try {
                if (comando == "addSong") {
                    cin >> c >> a >> d;
                    ipud.addSong(c, a, d);
                }
                else if (comando == "addToPlaylist") {
                    cin >> c;
                    ipud.addToPlaylist(c);
                }
                else if (comando == "current") {
                    ipud.current();
                }
                else if (comando == "play") {
                    pair <bool, string> sol = ipud.play();

                    if (sol.first) {
                        cout << "Sonando " << sol.second << endl;
                    }
                    else {
                        cout << sol.second << endl;
                    }
                }
                else if (comando == "totalTime") {
                    int t = ipud.totalTime();
                    cout << "Tiempo total " << t << endl;
                }
                else if (comando == "recent") {
                    cin >> n;
                    list<string> resultado = ipud.reciente(n);
                }
            }
            catch (...) {
                return false;
            }
        }
    }
}

```

```

        if (!resultado.empty()) {
            cout << "Las " << resultado.size() << " mas recientes" << endl;

            auto it = resultado.cbegin();

            while (it != resultado.cend()) {
                cout << "    " << *it << endl;
                it++;
            }
        }
        else {
            cout << "No hay canciones recientes" << endl;
        }
    }
    else if (comando == "deleteSong") {
        cin >> c;
        ipud.deleteSong(c);
    }
}

catch (invalid_argument& e) {
    cout << "ERROR " << e.what() << endl;
}

cin >> comando;
}

cout << "---" << endl;

return true;
}
}

int main() {
    // ajustes para que cin extraiga directamente de un fichero
#ifndef DOMJUDGE
    std::ifstream in("datos.txt");
    auto cinbuf = std::cin.rdbuf(in.rdbuf());
#endif

    while (resuelveCaso());

    // para dejar todo como estaba al principio
#ifndef DOMJUDGE
    std::cin.rdbuf(cinbuf);
    system("PAUSE");
#endif
    return 0;
}

```