

MAIN.CPP

```
include "sr2k8.h"
#include "MainWindow.h"

#define SCHEMADIALECT <PostgreSQL/PostgreSQLSchema.h>
#include <Sql/sch_source.h>

bool ConnectToPSQL();
void LoadConfig();

GUI_APP_MAIN
{
    ::SetLanguage( ::GetSystemLNG() );

    LoadConfig();
    if (ConnectToPSQL())
    {
        MainWindow mw;
        mw.Run();
    }
}

void LoadConfig()
{
    String cfgfile = ConfigFile();
    if (FileExists(cfgfile))
    {
        VectorMap<String, String> cfg = LoadIniFile(cfgfile);
    }
    else
    {
        // configuracion por default
        String s;
        s << "APPV=1.0\n"
        << "HOST=10.62.200.1\n"
        << "DATABASE=S2K8\n"
        << "USER=sis\n"
        << "PASS=sis\n";

        if (!SaveFile(ConfigFile(),s))
        {
```

```

    exit(-1);
}

}
}

bool ConnectToPSQL()
{
    if (!G_STATE->GetPsqlSession()->Open("host=localhost user=sis password=sis
dbname=S2K8"))
    {
        Exclamation(Format("Error abriendo base de datos: %s",
        DeQtf(G_STATE->GetPsqlSession()->GetLastError())));
        return false;
    }

    return true;
}

```

Sr2k8.H

```

#ifndef _sr2k8_sr2k8_h
#define _sr2k8_sr2k8_h

#include <CtrlLib/CtrlLib.h>
#include <SqlCtrl/SqlCtrl.h>

using namespace Upp;

// definiciones para PostgreSQL
#include <PostgreSQL/PostgreSQL.h>
#define SCHEMADIALECT <PostgreSQL/PostgreSQLSchema.h>
#define MODEL <sr2k8/sr2k8db.sch>
#include <Sql/sch_header.h>

#include "globalState.h"
#define G_STATE GlobalState::getInst()

// layout de las ventanas
#define LAYOUTFILE <sr2k8/sr2k8.lay>
#include <CtrlCore/lay.h>

#endif

```

globalState.h (singleton class spec)

```

#ifndef _sr2k8_globalState_h_
#define _sr2k8_globalState_h_

#include <PostgreSQL/PostgreSQL.h>

using namespace Upp;

// clase singleton que mantiene un estado global del sistema

class GlobalState
{
    static GlobalState* pInstance;

    PostgreSQLSession pgsq;
    String currentUser;

public:
    static GlobalState* getInst()
    {
        if (!pInstance)
            pInstance = new GlobalState;

        return pInstance;
    }

    PostgreSQLSession* GetPsqlSession() { return &pgsq; }

protected:
    GlobalState() {
        currentUser == "";
    };
    GlobalState(const GlobalState&);
    GlobalState& operator= (const GlobalState&);
};

#endif

```

This is initialized in globalstate.cpp as follows:

```

#include "globalState.h"

GlobalState* GlobalState::pInstance = NULL;

```

Do you see any problem related to the singleton??? Maybe it's that 'new Globalstate' on singleton

getInst() method.
