

```

#include "mainwindow.h"

#include <QApplication>

int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    MainWindow w;
    w.show();
    return a.exec();
}

#ifndef MAINWINDOW_H
#define MAINWINDOW_H

#include <QMainWindow>
#include <QString>
#include <QComboBox>
#include <QListWidget>
#include <QLabel>
#include <QMap>
#include <QStringList>

class MainWindow : public QMainWindow
{
    Q_OBJECT

public:
    MainWindow(QWidget *parent = nullptr);

private slots:

```

```
void guardarSeleccion();

void eliminarSeleccion();

void solicitarAlmacen();

void pagar();

void filtrarMedidasPorRin(int index);


private:

    QComboBox *marcaCombo;

    QComboBox *tipoCombo;

    QComboBox *rinCombo;

    QComboBox *medidaCombo;


    QListWidget *listaSeleccion;


    QLabel *subtotalLabel;

    QLabel *totalLabel;


    double subtotal;

    double total;


    QMap<QString, double> preciosMarca;


    void actualizarTotales();


    QStringList todasMedidas;

};


#endif // MAINWINDOW_H
```

```
#include "mainwindow.h"

#include <QVBoxLayout>
#include <QHBoxLayout>
#include <QLabel>
#include <QDateTime>
#include <QFile>
#include <QTextStream>
#include <QMessageBox>
#include <QComboBox>
#include <QPushButton>
#include <QListWidget>
#include <QDebug>
```

```
MainWindow::MainWindow(QWidget *parent)
    : QMainWindow(parent), subtotal(0), total(0)
```

```
{
    preciosMarca = {
        {"Pirelli", 500.0},
        {"Michelin", 600.0},
        {"Goodyear", 550.0},
        {"Bridgestone", 580.0},
        {"Firestone", 530.0}
    };
}
```

```
todasMedidas = {
    "185/65 R13",
    "195/60 R14",
    "205/55 R14",
    "215/60 R15",
}
```

```
"225/50 R16",  
"235/55 R17",  
"245/45 R18"  
};
```

```
QWidget *central = new QWidget(this);  
QHBoxLayout *mainLayout = new QHBoxLayout(central);
```

```
// Columna izquierda
```

```
QVBoxLayout *leftLayout = new QVBoxLayout();  
leftLayout->addWidget(new QLabel("MARCA"));  
marcaCombo = new QComboBox();  
marcaCombo->addItems(preciosMarca.keys());  
leftLayout->addWidget(marcaCombo);
```

```
leftLayout->addWidget(new QLabel("TIPO DE LLANTAS"));  
tipoCombo = new QComboBox();  
tipoCombo->addItems({"Carga", "Turismo", "Off-road", "4x4"});  
leftLayout->addWidget(tipoCombo);
```

```
leftLayout->addWidget(new QLabel("RIN"));  
rinCombo = new QComboBox();  
rinCombo->addItems({"13", "14", "15", "16", "17", "18"});  
leftLayout->addWidget(rinCombo);
```

```
leftLayout->addWidget(new QLabel("MEDIDA"));  
medidaCombo = new QComboBox();  
medidaCombo->addItems(todasMedidas);  
leftLayout->addWidget(medidaCombo);
```

```

QPushButton *cerrarSesionBtn = new QPushButton("CERRAR SESIÓN");
connect(cerrarSesionBtn, &QPushButton::clicked, this, &MainWindow::close);
leftLayout->addWidget(cerrarSesionBtn);

// Centro
QVBoxLayout *centerLayout = new QVBoxLayout();
QPushButton *btnSiguiente = new QPushButton("SIGUIENTE");
connect(btnSiguiente, &QPushButton::clicked, this, &MainWindow::guardarSeleccion);
centerLayout->addWidget(btnSiguiente);

// Derecha
QVBoxLayout *rightLayout = new QVBoxLayout();
listaSeleccion = new QListWidget();
rightLayout->addWidget(new QLabel("SELECCIONES"));
rightLayout->addWidget(listaSeleccion);

QLabel *fechaHora = new QLabel(QDateTime::currentDateTime().toString("dd MMM
yyyy\\nhh:mm AP"));
rightLayout->addWidget(fechaHora);

QPushButton *btnSolicitar = new QPushButton("SOLICITAR AL DE ALMACEN");
QPushButton *btnEliminar = new QPushButton("ELIMINAR");
QPushButton *btnPagar = new QPushButton("PAGAR");

subtotalLabel = new QLabel("SUBTOTAL: $0.00");
totalLabel = new QLabel("TOTAL: $0.00");

rightLayout->addWidget(btnSolicitar);

```

```

rightLayout->addWidget(subtotalLabel);
rightLayout->addWidget(totalLabel);
rightLayout->addWidget(btnEliminar);
rightLayout->addWidget(btnPagar);

connect(btnEliminar, &QPushButton::clicked, this, &MainWindow::eliminarSeleccion);
connect(btnSolicitar, &QPushButton::clicked, this, &MainWindow::solicitarAlmacen);
connect(btnPagar, &QPushButton::clicked, this, &MainWindow::pagar);
connect(rinCombo, QOverload<int>::of(&QComboBox::currentIndexChanged),
        this, &MainWindow::filtrarMedidasPorRin);

// Layout final
mainLayout->addLayout(leftLayout);
mainLayout->addLayout(centerLayout);
mainLayout->addLayout(rightLayout);
setCentralWidget(central);
}

void MainWindow::guardarSeleccion()
{
    QString marca = marcaCombo->currentText();
    QString tipo = tipoCombo->currentText();
    QString rin = rinCombo->currentText();
    QString medida = medidaCombo->currentText();

    QString itemText = marca + " - " + tipo + " - Rin " + rin + " - " + medida;
    listaSeleccion->addItem(itemText);

    double precio = preciosMarca.value(marca, 0);

```

```

subtotal += precio;

total = subtotal;

actualizarTotales();

QFile file("seleccion_llantas.txt");
if (file.open(QIODevice::Append | QIODevice::Text)) {
    QTextStream out(&file);
    out << "----- SELECCIÓN DE LLANTAS -----\\n";
    out << "Fecha: " << QDateTime::currentDateTime().toString("dd/MM/yyyy hh:mm AP") <<
"\\n";
    out << "Marca: " << marca << "\\n";
    out << "Tipo: " << tipo << "\\n";
    out << "Rin: " << rin << "\\n";
    out << "Medida: " << medida << "\\n";
    out << "Precio: $" << precio << "\\n";
    out << "-----\\n\\n";
    file.close();
    QMessageBox::information(this, "Guardado", "¡Selección guardada con éxito!");
} else {
    QMessageBox::warning(this, "Error", "No se pudo guardar la selección.");
}
}

void MainWindow::eliminarSeleccion()
{
    QListWidgetItem *item = listaSeleccion->currentItem();
    if (!item) {
        QMessageBox::warning(this, "Error", "Selecciona un ítem para eliminar.");
    }
}

```

```

        return;
    }

    QString texto = item->text();
    QString marca = texto.section(" - ", 0, 0);
    double precio = preciosMarca.value(marca, 0);

    subtotal -= precio;
    if (subtotal < 0) subtotal = 0;
    total = subtotal;

    actualizarTotales();
    delete item;
}

void MainWindow::solicitarAlmacen()
{
    if (listaSeleccion->count() == 0) {
        QMessageBox::information(this, "Solicitud", "No hay productos para solicitar.");
        return;
    }

    QString mensaje = "Solicitud enviada al almacén para los siguientes productos:\n";
    for (int i = 0; i < listaSeleccion->count(); ++i) {
        mensaje += listaSeleccion->item(i)->text() + "\n";
    }

    QMessageBox::information(this, "Solicitud", mensaje);
}

```



```

void MainWindow::pagar()
{
    if (listaSeleccion->count() == 0) {
        QMessageBox::information(this, "Pagar", "No hay productos en el carrito para pagar.");
        return;
    }

```

```

        QMessageBox::information(this, "Pago", QString("Se ha realizado el pago por un total de
        $%1").arg(total, 0, 'f', 2));
        listaSeleccion->clear();
        subtotal = 0;
        total = 0;
        actualizarTotales();
    }

```

```

void MainWindow::filtrarMedidasPorRin(int index)
{
    if (index < 0) return;
    QString rinSeleccionado = rinCombo->itemText(index);
    QString buscar = "R" + rinSeleccionado;

    medidaCombo->clear();
    for (const QString &medida : todasMedidas) {
        if (medida.contains(buscar)) {
            medidaCombo->addItem(medida);
        }
    }
}

```

```
if (medidaCombo->count() == 0) {  
    medidaCombo->addItem("No hay medidas para este rin");  
}  
}
```

```
void MainWindow::actualizarTotales()  
{  
    subtotalLabel->setText(QString("SUBTOTAL: $%1").arg(subtotal, 0, 'f', 2));  
    totalLabel->setText(QString("TOTAL: $%1").arg(total, 0, 'f', 2));  
}
```