

## Estudo Dirigido – Projeto em C#

O objetivo desta atividade é estimular a prática, de forma integrada, dos assuntos abordados durante todo o curso. A execução desta atividade pode ser de forma individual ou em dupla, sendo esta escolha a cargo de cada aluno. Recomenda-se que caso você sinta segurança, procure fazer individualmente ou com um colega que está sentindo dificuldades. Caso esteja em dupla, contribua com o colega nas dúvidas que ele tenha e permita que ele tente propor soluções para os problemas.

Você e sua dupla devem efetuar as atividades deste estudo de forma independente, procurando usar o conhecimento adquirido nas aulas e em seus estudos para os simulados. O professor irá auxiliar em casos de dúvidas, porém ao invés de respostas prontas, irá guiá-los para que vocês consigam descobrir quais métodos poderiam ser usados na situação em que a dúvida se manifestou.

Os exercícios podem ser realizados fora do horário da aula, mas o comparecimento nas aulas será obrigatório mesmo que seu projeto esteja terminado. Cada aula terá 4 horas, das quais será reservado um período entre 30 e 60 minutos no final da mesma para a resolução de questões com nível similar às da certificação.

### Cenário

Parabéns! Você foi contratado pela Líbero Rent a Car, uma pequena empresa de aluguel de carros, para desenvolver o sistema interno de controle de aluguel de carros da empresa.

Atualmente, a Líbero controla todo o fluxo em planilhas do Excel. Seu trabalho será modernizar este processo, desenvolvendo uma aplicação WPF se conectando a um WCF Data Service, que por sua vez utiliza um banco de dados para armazenar as informações. As seguintes planilhas são usadas atualmente, e devem ser substituídas por funcionalidades do seu sistema:

Planilha	Colunas
Carros	Modelo, Ano, Placa, Quilometragem
Clientes	Nome, CPF, Endereço, Dados do cartão de crédito
Aluguel	Carro, Cliente, Data de Aluguel, Data de Devolução, Km Inicial, Km Final

### Sistema Interno

Você deve implementar telas que permitam o cadastro de carros e clientes. Estas telas devem, além de listar todos os carros e todos os clientes já existentes, permitir que novos registros sejam adicionados.

O sistema deve possuir um controle de usuários com login e senha. A senha deve ser armazenada de forma segura, sem risco das mesmas serem expostas em caso de comprometimento do banco de dados.

Ao visualizar a listagem de carros, você deve indicar se este carro está atualmente alugado. Se estiver, você deve mostrar um botão para “Registrar Devolução”, que será usado quando o cliente devolver o carro. Se o carro não estiver alugado, você deve mostrar o botão “Alugar Carro”, usado para iniciar um aluguel.

Ao clicar no botão de Registrar Devolução, deve ser exibido para o usuário uma tela para informar a KM marcada no odômetro do carro no momento da devolução. Esta informação, além de ir para o registro do Aluguel, deve ser atualizada também no registro do carro. Deve também ser escrita uma linha no arquivo de log “activities.log”, indicando a data/hora atual, o usuário logado no sistema que registrou a devolução, a placa do carro e a quilometragem informada.

Ao clicar no botão de Alugar Carro, o usuário deve ser capaz de selecionar para qual cliente o carro está sendo alugado. Um registro do tipo Aluguel deve ser criado, com os campos devidamente preenchidos, quando um cliente

aluga um carro. Deve também ser escrita uma linha no arquivo de log “activities.log”, indicando a data/hora atual, o usuário logado no sistema que registrou o aluguel e a placa do carro.

A tela de listagem de carros também deve oferecer um filtro, com as seguintes opções: “Todos”, “Apenas Disponíveis” e “Apenas alugados”. Ao clicar nos filtros, devem ser exibidos para o usuário apenas os carros que respeitem estas restrições.

Como vários turistas do exterior procuram os serviços da empresa, é necessário exibir a cotação do dólar e do euro na tela inicial da aplicação. A API Cotações, disponível em <http://developers.agenciaideias.com.br/cotacoes> deve ser usada para obter estes valores. A aplicação não deve ficar travada enquanto espera a resposta da API.

Os dados de cartão de crédito dos clientes, por motivos óbvios, devem ser armazenados de forma segura.

O cadastro dos clientes deve permitir uma funcionalidade a mais que não está presente na planilha: Uma foto da carteira de habilitação deve ser gravada junto aos dados do cliente, sendo armazenada no campo FotoCNH do objeto que representa o cliente. Os arquivos das fotos podem chegar a 20MB de tamanho, logo deve-se preocupar com o tempo de acesso a disco durante a leitura e escrita destes arquivos.

O sistema deve ser agradável à vista do usuário, tendo um layout aceitável, com um menu constantemente exibido, com as opções “Carros”, “Clientes” e “Relatórios”.

Na geração de relatórios, devem existir dois itens:

1. Em um deles, deve ser criado um arquivo em formato .doc, contendo os dados de todos os carros cadastrados no sistema. Lembre-se que o processo de geração do relatório pode ser demorado, e sua aplicação não deve travar durante o processo de geração do documento. Você deve incluir uma barra de progresso indicando ao usuário quanto do relatório já foi gerado.
2. No outro item, deve ser exibido em tela a quantidade de alugueis efetuados para um carro em um determinado mês, assim como a média de quilômetros percorridos em cada aluguel. A tela deve oferecer ao usuário uma forma de selecionar o carro e o mês. Use uma abordagem assíncrona (com `async/await`) para o método que irá calcular estes valores, e **não** use o mecanismo de Dispatcher para atualizar a tela.

## Web Service

Os dados de clientes, carros e alugueis devem ser mantidos em um banco de dados. A aplicação cliente no entanto não deve ter acesso direto ao banco, e sim acessar os dados através de um WebService WCF Data Services.

O acesso aos dados deve ser feito através de propriedades que representam as coleções de entidades. A adição ou alteração da imagem da CNH deve ser realizada através de uma Service Operation. A adição, alteração e obtenção dos dados de cartão de crédito também deve ser realizada apenas através de Service Operations.

## BONUS 1

Usar criptografia de chave assimétrica para trafegar os dados da chamada das Service Operations que transmitem os dados de cartão de crédito.

## BONUS 2

Oferecer a opção de importação / exportação da listagem de carros (mesmos dados do relatório) a partir de / para um arquivo serializado de forma binária, além do .doc.

## BONUS 3

Atualizar os dados da cotação de dólar e euro a cada 5 minutos.