

PROJETO TESTE SOLÇÕES DE SOFTWARE PARA NEGCIOS

PT#09-SELETIVA WSC2017
Sessão 03



CONTEÚDO

Sessão 03 consiste dos seguintes documentos e arquivos:

- WireframeSessao03-E02.pptx
- bdAbuDhabiS03.sql
- bdAbuDhabiS03-MySQL.sql
- ImagensQuartos.zip
- XMLTemplate.zip

INTRODUÇÃO

Nesta sessão, você deverá continuar a desenvolver o Sistema AbuDhabi Tur. Você receberá todos os arquivos necessários para realizar o seu trabalho, independente de ter ou não realizado as etapas anteriores.

INSTRUÇÕES PARA COMPETIDOR

Ao final desta sessão, você precisará fazer suas entregas finalizadas e enviar dentro do tempo determinado (3 horas).

Quando solicitado cadastro, alteração, exclusão, pesquisas e importações, as entregas devem estar vinculadas diretamente ao banco e arquivos disponibilizados.

Sempre usar a credencial a seguir para autenticação nos bancos de dados envolvidos.

- Usuário: sa
- Senha: seletiva.123
-

O nome do banco de dados, quando usado, será definido nos scripts fornecidos dos datafiles.

ENTREGAS

3.1 IMPLEMENTAÇÃO DE MÓDULO DE RESERVAS COM UTILIZAÇÃO DO PADRÃO DE PROJETO

Nessa entrega você irá desenvolver um módulo onde será possível realizar o registro dos clientes nos subsistemas de reservas de hotel e reservas de passagens aéreas:

3.1.1 - DLI/Jar AgenciaReservas

Seguir a estrutura de classes disponibilizada no diagrama de Classes UML abaixo:

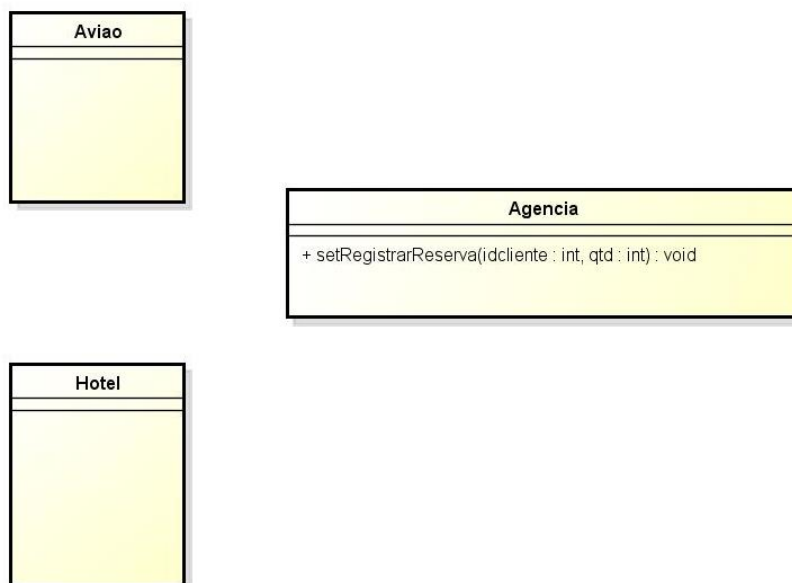


Figura 01 – Ilustração parcial da implementação do padrão de projeto usado.

Características:

- Ao invocar o método setRegistrarReserva da classe Agencia o mesmo deve invocar os métodos das classes Aviao e Hotel. Cada classe notificada faz a reserva, que é completada por meio da geração de um arquivo xml salvo na pasta do projeto.
- A função da classe agência é somente notificar a companhia de avião e hotel, que precisam fazer uma reserva para o cliente x (idcliente), quantidade de pessoas y (qtd)
- Todas as classes e método relacionado à problemática devem ser públicos
- As classes Hotel e Aviao devem conter o método setRegistrarReservas com a mesma assinatura de agencia.
- A classe Hotel irá adicionar em seu arquivo xml a informação do código e quantidade de pessoas, informado no método setRegistrarReservas. Não deve apagar informações já existentes no xml.
- A classe Aviao irá adicionar em seu arquivo xml a informação do código e quantidade de pessoas informado no método setRegistrarReservas. Não deve apagar informações já existentes no xml.
- Os arquivos xml deverão estar no mesmo diretório que encontra-se a sua dll/jar

Insumos

Serão disponibilizados os arquivos XML modelo para armazenamento dos dados;

3.12 – Método gerador de classes

Na classe **agencia** criar método público com a assinatura gerarClasse(String nomeClasse, String[] atributos).

Ao executar, o método deve gerar um txt (minhaClasse.txt) com a estrutura de uma classe de acordo

com os parâmetros, sendo: nomeClasse (nome da classe), atributos (os atributos da classe, onde seu tipo é definido pela primeira letra – vide Tabela 01 - Exemplo).

A estrutura gerada no txt deve representar fielmente uma classe, seguindo seus padrões, de forma que se for copiada em um projeto externo, sua sintaxe a estrutura da linguagem garanta seu funcionamento para compilação, a mesma deve estar “identada”.

Os atributos devem estar encapsulados e limita-se aos tipos a seguir: string, inteiro, double e float, boolean).

Nome do Atributo informado	Inicial	Tipo	Nome do Atributo Gerado
ildade	i	Inteiro	idade
sNome	s	String	nome
dValor	d	Double	valor
fDinheiro	f	Float	<u>dinheiro</u>
bExiste	B	Boolean	existe

Tabela 01 - Exemplo

Arquivos para serem entregues e convenções para essa entrega

- AgenciaReservas (dll ou jar)
- AgenciaReservas (Pasta com Projeto – CódigoFonte)

3.2 MODULO PARA REALIZAÇÃO DE CRUD DE APARTAMENTOS

Criar janela/tela com base no arquivo WireframeSessao03-E02.ppsx

Usar o script bdAbuDhabiS03.sql/ bdAbuDhabiS03-MySQL.sql A estrutura de definição de dados do banco de dados não pode ser alterada.

- **Tela Listagem de Apartamentos**

1. O sistema deverá dispor os apartamentos em uma matriz, com 10 colunas e n linhas;
2. Para cada apartamento deve ser mostrado o ícone do tipo, o código, uma cor de fundo relacionada a situação do apartamento (ativo – verde, inativo – vermelho, manutenção – roxo (neste caso, mostrar também a data de liberação do apartamento));
3. Para apartamentos em manutenção, ao clicar com o botão direito do mouse o sistema deve exibir a opção **Histórico de Manutenção** que ao ser clicada deverá exibir o histórico de manutenções do apartamento selecionado com base no banco de dados entregue.
4. Deve conter uma área onde sejam exibidos os tipos de apartamentos com suas respectivas imagens. A medida que o usuário for selecionando por meio de um clique, o sistema exibirá os apartamentos correspondentes aos tipos selecionados.
5. Para desfazer a seleção clique novamente no tipo.
6. Destacar tipos selecionados, diferenciando cor de fundo.
7. Deve conter um botão para adicionar apartamentos, que ao ser clicado abrirá a tela Cadastrar Apartamentos.
8. Ao clicar com o botão direito do mouse em qualquer apartamento, o sistema deverá exibir, uma opção de exclusão (deletar), que ao ser clicado uma mensagem de confirmação será exibido. Confirmado deverá excluir o apartamento selecionado, caso contrário, retorna tela anterior.

- **Tela Cadastro de Apartamento**

1. **Campo Tipo** - Deve ser possível selecionar um tipo clicando no mesmo
2. **Campo Fotos** - Deve ser possível adicionar e remover várias imagens do apartamento. Ao adicionar uma imagem (exibir opção para escolher imagens armazenadas no computador) esta deve ser exibida na área fotos. Sobre cada foto deverá existir um botão para removê-la.
3. **Campo Itens Adicionais** – O usuário poderá adicionar diversos objetos ao apartamento, para isso, basta marcar a caixinha de confirmação do item.
4. **Campo Código** – O código deve ser gerado automaticamente após o usuário selecionar o tipo. Ex: **A09AP01**
 - (a) A código em verde significa o número do andar em que o apartamento vai ser cadastrado (formado por dois dígitos, sempre);
 - (b) As letras em vermelho são as duas primeiras letras do tipo do apartamento;
 - (c) Por fim, o número final é a posição sequencial em que o apartamento do tipo ocupa no andar; atenção: a sequência é em relação ao tipo de apartamento no andar. Ex: se já existe um apartamento do tipo Chalé no andar 2 o código dele é A02CH01. Porém se adicionarmos outro apartamento só que do tipo Americano no mesmo andar 2, o código deste ficará A02AM01;
5. **Campo Situação** – Neste momento o usuário deverá clicar em um dos três botões existentes:
Inicialmente os botões devem estar na cor cinza, ao clicar em um dos botões, este deve mudar para a sua cor correspondente.
 - (a) Ativo – Cor verde
 - (b) Inativo – Cor vermelho
 - (c) Manutenção – Cor Roxa (Ao clicar em manutenção o sistema deverá exibir uma tela onde seja possível selecionar a data de início e fim da manutenção (tabela existente

no banco de dados)).

6. **Campo Leitos** – Onde será informado a quantidade máxima de pessoas no apartamento;
 7. **Campo Qtd. Quartos** – Quantidade de Quartos existentes no apartamento;
 8. **Proprietário** – Selecionar o proprietário do apartamento (tabela existente no banco de dados);
 9. **Responsável** – Selecionar o responsável pelo apartamento (tabela existente no banco de dados);
 10. **Vídeo** – ao dar um duplo clique na área do vídeo deve ser aberto a opção para escolher um vídeo avi. Após a escolha o vídeo é carregado em sua área de reprodução. Quando clicar em play vídeo deve ser reproduzido do início. Para parar clicar no botão stop (quadrado.). Ao salvar o novo apartamento o vídeo deve persistir diretamente na base de dados.
- **Tela Editar Apartamentos**
 1. Onde será possível alterar os dados de todos os campos descrito na **tela de cadastro**, com exceção do código;
 2. A tela de Edição deve ser acionada quando o usuário der dois cliques encima de qualquer apartamento atualmente exibido na tela;

Arquivos para serem entregues e convenções

- Sessa03Apartamento.exe

3.3 CONSULTA EM BANCO DE DADOS

Utilizando o mesmo banco de dados disponibilizado na entrega anterior. Você deverá criar um script (selectApartamentoS03E03.sql) com **select** que retorne os dados de acordo com a estrutura descrita abaixo:

Apartamento	2016	2015	2014	2013	Media Dias
A02CH01	0	142	159	116	104,25
A02CH02	3	245	146	169	140
A02AP01	0	100	80	125	76,75
A02CA01	0	98	112	33	60,75
A09CH01	0	5	40	77	30,5
A04AP05	4	33	9	12	14,25
A04AM05	0	30	126	100	64
A07EU05	0	45	10	45	25,5
A01AP06	1	96	14	78	47

O relatório deve listar para cada apartamento e para os 4 últimos anos (incluindo o ano atual), a quantidade de dias em que este ficou em manutenção e ainda deve mostrar a média de todos os anos;

OBS* Os anos devem estar dispostos em 4 colunas diferentes de forma decrescente; (Seguir modelo da imagem acima);

Ordene os dados de forma decrescente pela última coluna (Media Dias).

Para que o teste funcione adequadamente o seu arquivo script só deve conter o select necessário para retorno dos dados.

Arquivos para serem entregues e convenções

- selectApartamentoS03.sql