



LOL - Lavanderia On-Line

UFPR - SEPT - TADS
Desenvolvimento WEB-II

ENUNCIADO

Vocês deverão desenvolver um sistema de Controle de Lavanderia chamado LOL - Lavanderia On-Line. Esse sistema controla pedidos de lavagem de roupas.

O acesso ao sistema é feito por meio de 2 perfis: Cliente e Funcionário. Todas as funcionalidades necessitam login no sistema, exceto o Autocadastro de clientes e o próprio Login.

- **RF000 - Autocadastro:** Qualquer pessoa pode se cadastrar no sistema LOL, indicando : CPF (único), Nome, E-mail (único), Endereço completo, Telefone. O login será o e-mail da pessoa, mais uma senha aleatória que será enviada por e-mails no ato do autocadastro.
- **RF001 - Login:** Qualquer pessoa cadastrada (qualquer perfil) pode fazer login no sistema informando seu e-mail e senha. O sistema automaticamente identifica o perfil do usuário.

PERFIL CLIENTE

- **RF002 - Página Inicial de Cliente:** Imediatamente após o login deve ser mostrada uma lista com todos os pedidos em aberto do cliente.
- **RF003 - Pedido on-line:** O cliente faz o pedido de serviço de lavagem no sistema e, mais tarde, a empresa envia um *motoboy* para coletar as roupas. Devem ser discriminadas todas as roupas a serem lavadas e o sistema mostrará ao cliente o orçamento (valor e prazo) para o serviço. As roupas são lavadas no prazo da roupa mais demorada.
- **RF004 - Aprovar Orçamento do Pedido:** Após a apresentação do orçamento (valor e prazo), o cliente pode aprovar o pedido. Em sendo aprovado, deve ser dado ao cliente um número de pedido, que pode ser usado para consulta, e o pedido passa para o estado EM ABERTO.
- **RF005 - Rejeitar Orçamento do Pedido:** Após a apresentação do orçamento (valor e prazo), o cliente pode rejeitar o pedido. Sendo rejeitado o pedido passa para o estado REJEITADO.
- **RF006 - Listagem de pedidos:** Onde o cliente obtém uma lista de todos os seus pedidos. Deve ser disponibilizado um filtro para consultar somente os pedidos que estejam em um



LOL - Lavanderia On-Line

UFPR - SEPT - TADS
Desenvolvimento WEB-II

dos seus estados específicos. A listagem deve ser mostrada ordenada decrescentemente por data/hora. Deve ser disponibilizado um link/botão de ação para que o cliente possa pagar o pedido.

- **RF007 - Cancelamento de pedidos:** Se o motoboy ainda não passou recolher suas roupas, o cliente pode cancelar seu pedido, sendo que este vai para o estado CANCELADO.
- **RF008 - Consulta de pedido:** Deve ser disponibilizada uma tela em separado onde o cliente informa o número do pedido e são mostrados todos os dados deste: as roupas no pedido, preços, preço total, prazo do serviço e situação.
- **RF009 - Pagar pedido:** Assim que o pedido entrar no estado AGUARDANDO PAGAMENTO significa que ele já foi lavado e agora o cliente precisa efetuar o pagamento. Deve ser apresentada uma tela com os dados do pedido e um botão que simplesmente confirma o pagamento do pedido, registrado a data/hora.

PERFIL FUNCIONÁRIO

- **RF010 - Página Inicial de Funcionário:** Devem ser mostrados todos os pedidos EM ABERTO que ainda não foram recolhidos pelo motoboy, ordenados de forma crescente por data/hora. Também deve ser disponibilizado um link/botão para que o Funcionário possa registrar o recolhimento do pedido.
- **RF011 - Confirmação de recolhimento:** onde o funcionário informa que o motoboy recolheu as roupas do cliente. O sistema altera o estado do pedido automaticamente para RECOLHIDO. Esta funcionalidade deve ser fornecida de duas formas:
 - a. Via link proveniente da Página Inicial do Funcionário, e
 - b. Via link proveniente da Página de Visualização de pedidos.
- **RF012 - Visualização de pedidos:** O funcionário lista todos os pedidos, que podem ser filtrados como: pedidos de hoje, pedidos por período de datas (início e fim), ou todos os pedidos. Os pedidos devem ser mostrados em ordem crescente por data/hora. Deve ser usada a seguinte escala de cores:
 - a. Amarelo : pedido EM ABERTO



LOL - Lavanderia On-Line

UFPR - SEPT - TADS
Desenvolvimento WEB-II

- b. Vermelho : pedido REJEITADO ou CANCELADO
- c. Cinza : pedido RECOLHIDO
- d. Azul : pedido AGUARDANDO PAGAMENTO (já foi lavado)
- e. Alaranjado : pedido PAGO pelo cliente
- f. Verde : pedido FINALIZADO

Devem ser apresentados links/botões de ação para cada pedido em seu estado atual:

- a. EM ABERTO: botão/link para confirmação de recolhimento
 - b. RECOLHIDO: botão/link para confirmação de lavagem
 - c. PAGO: botão/link para finalização do pedido
- **RF013 - Confirmação de lavagem:** O funcionário registra que o pedido já foi lavado. O sistema altera automaticamente o estado para AGUARDANDO PAGAMENTO. Essa funcionalidade vem por meio de botão/link de ação na tela de visualização de pedidos.
 - **RF014 - Finalização de Pedido:** O funcionário registra que o pedido foi finalizado e as roupas enviadas ao cliente. O sistema automaticamente altera o estado para FINALIZADO. Essa funcionalidade vem por meio de botão/link de ação na tela de visualização de pedidos.
 - **RF015 - Manutenção de Peças de Roupas:** (CRUD - Inserção, Remoção, Atualização e Listagem) Um funcionário pode manter peças de roupas a serem lavadas. Por exemplo: calça jeans, camisa social, bermuda, etc. Deve ser armazenado também o preço individual e o prazo (em dias) para lavagem desta peça roupa;
 - **RF016 - Manutenção de funcionários:** (CRUD - Inserção, Remoção, Atualização e Listagem) O funcionário pode manter novos funcionários para acesso ao sistema, com os seguintes dados: e-mail único para login, nome, data de nascimento, e senha.
 - **RF017 - Relatório de Receitas em PDF:** O funcionário visualiza a receita conseguida pela empresa em um determinado período por meio de um filtro do relatório: data inicial e data final, que podem ser vazias. Os dados devem ser mostrados agrupados por dia.
 - **RF018 - Relatório de Clientes em PDF:** O funcionário visualiza os dados de todos os clientes.
 - **RF019 - Relatório de Clientes Fiéis em PDF:** O funcionário visualiza os dados dos cinco clientes que mais usam o serviço da lavanderia, contendo também a quantidade de



LOL - Lavanderia On-Line

UFPR - SEPT - TADS
Desenvolvimento WEB-II

pedidos e receita de cada um destes clientes;

- **RF020 - Relatório de Pedidos em PDF:** O funcionário visualiza os dados dos pedidos em um determinado período usando como filtro do relatório a data inicial e data final (que podem ser vazias), ordenados crescentemente por data/hora.

Toda e qualquer suposição, que não esteja definida aqui e que a equipe faça, deve ser devidamente documentada e entregue em um arquivo .doc/.odt que acompanha o trabalho.

DADOS INICIAIS PARA TESTES

Deve-se apresentar um conjunto mínimo de dados previamente cadastrados (pode-se apresentar mais):

- 2 funcionários (Maria e Mário)
- 3 clientes (João, José, Joana)
- 5 peças de roupas (calça, camisa, camiseta, meia e cueca)

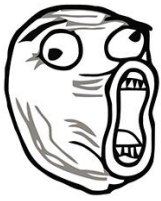
Deve-se prover os seguintes pedidos mínimos iniciais para teste (pode-se apresentar mais):

- **João:** Dois pedidos devem estar EM ABERTO. Um deve estar REJEITADO. Um deve estar CANCELADO. Um deve estar RECOLHIDO. Um deve estar FINALIZADO.
- **José:** Um pedido deve estar EM ABERTO. Um deve estar AGUARDANDO PAGAMENTO. Um deve estar PAGO. Dois pedidos devem estar FINALIZADOS.
- **Joana:** Um pedido deve estar RECOLHIDO. Dois devem estar AGUARDANDO PAGAMENTO. Um deve estar PAGO. Um deve estar FINALIZADO.

REQUISITOS NÃO-FUNCIONAIS

Os requisitos não-funcionais deste sistema são:

- Leiaute de telas deve ser bem elaborado;
- Deve ser usado DHTML (html/xhtml, css, dom e javascript);



LOL - Lavanderia On-Line

UFPR - SEPT - TADS
Desenvolvimento WEB-II

- Deve-se usar as tecnologias vistas em aula: Servlets, JSP, JSTL, EL. Bem como as boas práticas de programação: MVC, DAO, etc;
- Deve-se seguir boas práticas de programação e orientação a objetos: ocultamento de informações, baixo acoplamento, nomeação de atributos, classes e métodos, etc;
- Deve-se usar um *framework* para desenvolvimento das telas. Sugere-se o Bootstrap. Também deve-se usar um conjunto de bibliotecas Javascript para alterar o comportamento de telas, de forma dinâmica, quando necessário. Sugere-se o jQuery;
- Todos os campos devem possuir validação tanto em Javascript como em Java (Servlets), bem como máscaras;
- Todas as senhas devem ser criptografadas usando Hash SHA-256;
- Todas as tabelas no banco de dados devem estar normalizadas e devem seguir um padrão de codificação, inclusive as que não possuem cadastro e devem estar previamente preenchidas, exemplo: Cidade e Estado;
- Queries no banco de dados devem favorecer o desempenho por meio de JOINS e boas práticas de consultas;
- No caso da entrega do protótipo, todas as funcionalidades devem ser implementadas e todos os dados devem ser fictícios;
- O preenchimento do endereço do Cliente deve ser feito de forma automática consultando o CEP com a API Viacep (<https://viacep.com.br/>);
- Todas as datas e valores monetários devem ser entrados e mostrados no formato brasileiro;
- Todos os campos que tiverem formatação devem possuir máscara;
- Todas as datas deverão ser entradas através de calendários;
- Qualquer tipo de remoção deve ser confirmada antes de ocorrer;
- O sistema será testado usando o navegador FIREFOX, versão mais recente.