

Projeto 2022/2023 – Empresa de Autocarros

Descrição do Problema:

Uma empresa de autocarros “AoR - Autocarros” está a planear desenvolver uma nova aplicação para a gestão das requisições dos autocarros disponíveis e de operações usualmente efetuadas (consultas, reservas, pagamentos, etc).

Os utilizadores podem ser de dois tipos:

1. Administradores;
2. Clientes.

Todos os utilizadores devem autenticar-se através de um sistema de *login* com nome de utilizador e palavra-chave quando acedem à aplicação. Os clientes têm privilégios diferentes de acordo com o pacote de subscrição que tenham com a empresa de autocarros. Existem 2 níveis diferentes de subscrição: normal e premium. As diferenças entre eles são as seguintes:

1. **Normal:** Têm uma penalização de 50% sempre que cancelarem uma reserva até 7 dias consecutivos antes da partida. Após este prazo não existe reembolso. Esta subscrição não representa nenhum custo mensal acrescido.
2. **Premium:** Podem cancelar as reservas sem qualquer custo até 2 dias consecutivos antes da data da reserva. Após este prazo não existe reembolso. Os utilizadores premium têm prioridade na reserva, ou seja, sempre que um utilizador premium procurar uma reserva, apenas os autocarros de outros utilizadores premium devem ser considerados como reservados. Havendo reservas de utilizadores normais para as datas pretendidas, e mais nenhum autocarro disponível, deve ser cancelada a reserva mais recente do utilizador normal cujo autocarro satisfaz as necessidades do cliente premium, desde que este cancelamento seja no mínimo com 2 dias de antecedência. Quando uma reserva é cancelada, o cliente afetado deve receber uma notificação (mensagem no ecrã). Esta subscrição tem um custo mensal de 10€.

Todos os clientes podem reservar viagens, consultar e cancelar as suas reservas. Os administradores podem gerir os clientes, autocarros e os motoristas.

Perante um cenário em que para as datas escolhidas por um cliente, todos os autocarros estejam reservados, os clientes podem continuar a fazer a reserva ficando em lista de espera. Nestes casos, não são aceites os respetivos pagamentos ao processar a reserva para lista de espera. Quando uma reserva é cancelada, os clientes em lista de espera recebem uma notificação (mensagem no ecrã). A partir desse momento, deverá ser processado o pagamento da sua reserva. Caso exista mais que um cliente em lista de espera, a vaga é atribuída ao cliente que efetuar primeiro o pagamento.

A interação com o utilizador deverá ser realizada através de apropriadas janelas e botões de Interface gráfica, de modo a permitir a entrada de dados e a apresentação de resultados.

Requisitos funcionais do Sistema:

1. **Login** via e-mail e palavra-chave, e **logout**: todos os utilizadores (clientes e administradores) devem fazer *login*, especificando e-mail e palavra-chave, antes de aceder às funcionalidades do sistema, podendo terminar a sua utilização, efetuando *logout*;
2. **Alterar palavra-chave**: Todos os utilizadores devem poder alterar a sua palavra-chave;
3. **Pagamentos**: Ao finalizar a reserva, o cliente deve ter a possibilidade de escolher o modo de pagamento: **PayPal**, **cartão crédito**, ou por **multibanco**. No caso do PayPal o sistema deve pedir e-mail e *palavra-chave* da conta do PayPal. No caso do cartão crédito o sistema deve pedir o número do cartão, nome do cliente no cartão, data de expiração e código de segurança. No caso do pagamento por multibanco, o sistema deve gerar os valores para campos entidade, referência e montante e mostrar ao utilizador. Após inseridos os dados do método de pagamento escolhido corretamente, o sistema deve considerar a reserva como paga validando o preenchimento dos campos;
4. **Página de Registo**: A página inicial, para além da função de *login* de utilizadores já registados, deve permitir que novos clientes se registem de forma autónoma no sistema; cada cliente é caracterizado pelo nome, NIF, morada, telefone, e-mail, e uma palavra-chave definida pelo cliente; cada cliente tem ainda um ID único;

Funcionalidades dos Administradores:

1. **Adicionar outros administradores**: Os administradores são caracterizados pelo nome, NIF, morada, telefone, e-mail, e uma palavra-chave inicial. Cada administrador tem ainda um ID único;
2. **Adicionar/Editar/Remover Motoristas**: Os administradores devem poder adicionar/editar/remover motoristas no sistema. Os motoristas têm associado o seu nome e e-mail;
3. **Adicionar/Editar/Remover autocarros**: O sistema deve permitir ao administrador adicionar/editar/remover autocarros; um autocarro é caracterizado por matrícula, marca, modelo, e a sua lotação (número de lugares); quando um autocarro é removido todas as reservas a ele associadas devem ser canceladas, automaticamente, havendo o respetivo reembolso para os clientes afetados de acordo com o plano de subscrição do cliente. O **reembolso** é sempre efetuado pelo mesmo método de pagamento utilizado no ato da reserva, sendo que no caso de pagamento através de referência multibanco será solicitado por um email o IBAN para processar o reembolso do devido valor.
4. **Adicionar/Editar/Remover Clientes**: Os administradores devem ter a possibilidade de adicionar/editar/remover os clientes; cada cliente é

caracterizados pelo nome, NIF, morada, telefone, e-mail, e uma palavra-chave; esta palavra-chave deve ser definida automaticamente quando um cliente é adicionado pelo administrador, tendo o cliente a possibilidade de a alterar depois na sua página; cada cliente tem ainda um ID único; quando um cliente é removido, todas as suas reservas devem ser canceladas automaticamente, havendo o respetivo reembolso de acordo com o plano de subscrição do cliente e o método de pagamento utilizado. Quando um cliente é removido, ele continua a conseguir fazer login na sua conta, de modo a conseguir saber que as suas reservas foram canceladas, no entanto, não consegue efetuar nenhuma operação de reserva.

5. **Listar todos os motoristas e clientes;**
6. **Visualizar estatísticas:**
 - a. Total de clientes;
 - b. Total de motoristas;
 - c. Total de autocarros;
 - d. Identificar o autocarro mais requisitado;
 - e. Identificar o cliente com mais viagens;
 - f. Listar todos os autocarros reservados e a respetiva data num determinado mês;
 - g. Listar as requisições canceladas num determinado mês;
 - h. Listar os clientes/reservas em espera;
 - i. Volume de reservas em cada mês e o dia do ano em que houve mais reservas.
7. O **grupo de 3 pessoas** deve implementar um menu que permita ao administrador gerir os seus motoristas com campos extra. Assim devem ser criadas as seguintes funcionalidades extra:
 - a. Registo de um salário base e de um valor cobrado por hora de serviço. Estes valores podem ser editados pelo administrador e podem variar entre os motoristas.
 - b. Registo de horas semanais de serviço de cada motorista;
 - c. Com base nos valores do ponto a) e das horas do ponto b), calcular o salário mensal de cada motorista.
 - d. Consulta do histórico de salários mensais de cada motorista.
 - e. Associação dos motoristas aos autocarros disponíveis, indicando quais os autocarros é que um determinado motorista pode conduzir. No momento da reserva de um serviço (autocarro) por parte de um cliente, deve ser alocado ao autocarro designado para fazer o serviço um dos motoristas autorizado a conduzir o mesmo. Se um autocarro estiver disponível, mas nenhum dos motoristas livres tiver autorização para o conduzir, então este autocarro não deve ser considerado disponível.
 - f. Consulta do histórico de serviços de cada motorista fez, com todas as informações relativas à reserva.
 - g. Consulta das reservas de serviços pendentes para cada motorista, com todas as informações relativas à reserva.

Funcionalidades dos Clientes:

1. **Realizar reserva de autocarro:** Um cliente deve conseguir efetuar reserva de um autocarro. Para isso, o cliente deve indicar a data do aluguer (que pode ser um ou vários dias), o número de pessoas a transportar, a partida, o destino, e o número de quilómetros esperados a realizar. De acordo com esta informação, o sistema realiza a reserva do autocarro disponível que minimiza o número de lugares vazios, indicando ao utilizador o autocarro a que ele terá direito. Para além disto, é ainda necessário alocar um único motorista a cada viagem, independente da sua duração. Esta atribuição é feita pelo sistema, alocando um motorista disponível. O custo da viagem é calculado da seguinte forma: $\text{Quilómetros (a efetuar)} * 0.55 + 1.2 * \text{Número de Pessoas}$; O grupo de 3 pessoas deve implementar um menu para os administradores que permita editar o custo de quilómetros e a taxa por pessoas associadas a cada autocarro.
2. **Ver histórico de reservas:** O cliente pode ver todo o histórico das reservas anteriormente efetuadas;
3. **Consultar reservas:** Os clientes podem consultar a lista de reservas que têm agendadas e as que eventualmente possam estar em lista de espera;
4. **Cancelar reservas:** Os clientes podem cancelar as reservas que têm. Aqui devem ser considerados os prazos e as penalizações do pacote de subscrição desse cliente. Quando é efetuada um cancelamento, o cliente recebe um aviso sobre a penalização a que está sujeito;
5. **Subscrição:** Os clientes podem gerir a sua subscrição (normal ou premium) a qualquer momento, não havendo devoluções de dinheiro no caso de um utilizador que cancele a sua subscrição premium.

Implementação:

A aplicação deve ser implementada na linguagem Java e deverá ter em conta os seguintes aspetos:

1. Ao entrar na aplicação, devem persistir todos os dados que se encontravam no fim da última utilização;
2. Cada classe deve gerir internamente os seus dados, pelo que deverá cuidar da proteção das suas variáveis e métodos;
3. Devem ser usados ficheiros de objetos para guardar os dados relativos às classes criadas;
4. Não deverão ser criados getters ou setters desnecessários.
5. Cada objeto deverá ser responsável por uma tarefa ou objetivo específico, não lhe devendo ser atribuídas funções indevidas;
6. Utilize as *keywords* `static` e `break` apenas quando tal se justifique e não para contornar possíveis erros do compilador ou falhas dos algoritmos desenvolvidos.

Elabore um diagrama com as suas classes e objetos (em UML), antes de iniciar a implementação, para prever a estrutura do projeto.

Tenha ainda em conta os seguintes pontos que serão importantes na **avaliação**:

1. Comentar as classes, métodos e variáveis públicas segundo o formato Javadoc.
2. Isto permitir-lhe-á gerar automaticamente uma estrutura de ficheiros HTML, descritivos do seu código, que deve incluir no seu relatório;
3. Comentar o restante código sempre que a leitura dos algoritmos não seja óbvia;
4. Tal como sugerido acima, evitar o uso abusivo de *static*, *break* e de variáveis e métodos *public*;
5. Na escolha de nomes para variáveis, classes e métodos, seguir as convenções adotadas na linguagem Java;
6. Na organização das classes deverá ser evitada a redundância dos dados.

Prazos de Entrega:

A entrega do trabalho compreende duas metas distintas:

1. **Meta 1:** Entrega até **12 de dezembro (23H59)** – Os alunos devem entregar um relatório PDF através do InforEstudante que deve incluir os diagramas de classe UML e o *design* inicial da interface gráfica da aplicação;
2. **Meta 2:** Entrega até **10 de janeiro (23H59)** – Os alunos devem submeter um ficheiro .zip através do InforEstudante que irá corresponder à versão final da aplicação. Este ficheiro .zip deverá incluir:
 - a. Código fonte de todas as classes Java;
 - b. Ficheiros executáveis;
 - c. **Ficheiros para teste da aplicação com valores iniciais;**
 - d. Ficheiros Javadoc;
 - e. Um relatório em PDF com os seguintes elementos:
 - i. Descrição geral da estrutura da aplicação;
 - ii. Diagrama de Classes;
 - iii. Descrição das principais estruturas de dados e dos ficheiros usados;
 - iv. Breve explicação de como é que as principais partes da aplicação estão implementadas e interligadas;
 - v. *Design* da interface gráfica da aplicação;

Os trabalhos serão comparados (tanto entre os trabalhos da disciplina como com código disponível na Internet), no sentido de detetar eventuais fraudes por cópia. Nos casos em que se verifique que houve cópia de trabalho total ou parcial, os grupos envolvidos terão os projetos anulados, reprovando à disciplina.

Composição dos grupos:

O trabalho deve ser realizado em grupos de **2 elementos**, com a exceção de 1 grupo de 3 elementos. A defesa dos trabalhos em cada uma das metas é feita individualmente, bem como a respetiva avaliação.

Avaliação do trabalho:

Para a avaliação do trabalho contam fatores de dois tipos:

1. Caixa preta (tal como é percecionado pelo utilizador):
 - a. Conjunto de funcionalidades implementadas;
 - b. Robustez do programa;
 - c. Qualidade da interface.
2. Caixa branca (a forma como está construído):
 - a. Qualidade das soluções técnicas encontradas para os problemas em causa;
 - b. Estruturação do código;
 - c. Validação de dados;
 - d. Qualidade dos comentários.

Nota: Não se aceitam trabalhos que apresentem erros de compilação no momento da defesa e que não estejam corretamente estruturados do ponto de vista da Programação Orientada aos Objetos.

Defesa final do trabalho

O trabalho deve ser defendido através de uma discussão presencial e individual. Para isso, cada grupo deve inscrever-se num horário que esteja disponível para essa defesa no InforEstudante. **As defesas irão ocorrer no dia 13 de janeiro de 2023.**