

Base de Dados
Coimbra Bus
Projeto - DEEC

Leonardo Gonçalves-2020228071
Gonçalo Bastos - 2020238997

Maio 2023

1 Introdução

Este relatório apresenta o projeto "Coimbra Bus", uma aplicação de bases de dados desenvolvida em Python para a empresa de transportes Coimbra Bus (CB). O objetivo deste projeto foi criar um sistema que permita a gestão de clientes, viagens, autocarros e operações típicas do setor de transportes, como consultas, reservas e pagamentos. A aplicação permite que os clientes consultem viagens disponíveis para determinado destino e dia, façam reservas, cancelem reservas e recebam mensagens enviadas pelos administradores. Além disso, a aplicação permite que os administradores gerenciem o sistema, adicionando novas viagens e realizando outras tarefas de gestão. Aqui será apresentada a organização geral da aplicação, alguns screenshots ilustrativos do aspeto geral da aplicação e das suas partes mais importantes, o diagrama Entidade-Relacionamento (ER) e o diagrama físico simplificado no ONDA, a distribuição final de tarefas por cada elemento do grupo, o tempo gasto com a disciplina por semana e por aluno desde o início do semestre, e outros aspetos relevantes. O relatório será concluído com uma síntese das principais conclusões do projeto.

2 Organização geral da aplicação

A nossa aplicação corre toda do '**main.py**' de onde é apresentado o menu principal onde o user pode fazer login como cliente ou como administrador, ou se for um cliente novo registar-se como tal. As funções para este propósito estão implementadas no ficheiro '**pag_inic.py**', sendo elas:

- **login()**, pede as credenciais ao user e verifica se existe na base de dados, caso exista, retorna o nif e uma flag que identifica se é cliente ou administradores.
- **register_cliente()**, pede os dados pessoais ao cliente, necessarios para o registo, faz a verificação dos mesmos, caso esteja tudo ok, adiciona o cliente a base de dados e retorna True.

Feito o registo o cliente pode fazer login, fazendo login sendo cliente é chamada a função **cliente_menu(nif)**, esta função esta implementada no ficheiro **cliente.py**. No menu cliente são apresentadas as opções de:

- Consultar Viagens
- Consultar Reservas
- Mensagens
- Dados Pessoais
- Logout

Se for administrador é chamada a função **admin_menu(nif)** implementada no ficheiro **admin.py**.

No menu de administrador são apresentadas as opções:

- Gerir Viagens
- Gerir Clientes
- Gerir Autocarros
- Enviar Mensagens
- Ver Estatísticas
- Logout

Estes menus funcionam com base em funções implementadas nos ficheiros:

- **gerir_viagens.py**, tem a implementação de funções que permitem ao administrador adicionar, eliminar ou alterar uma viagem, e também para o cliente Listar todos os destinos (permitir filtrar por parte do nome, distância de Coimbra e duração da viagem), listar todas as viagens disponíveis para um destino (tanto de ida como de regresso), dada uma gama de datas e ainda pesquisar viagens usando diversos critérios.

- **gerir_clientes.py**, tem implementadas funções que permitem ao administrador gerir os clientes, isto é, adicionar, eliminar e ainda atribuir7remove estatuto gold().
- **gerir_reservas.py**, implementação de funções que permitem ao cliente reservar um lugar numa viagem, ou ficar em fila de espera caso esteja cheio e listar todas as viagens para que tem reservas no futuro, bem como todas as que fez no passado.
- **gerir_autocarros.py**, implementação das funções que permitem ao administrador visualizar e adicionar ou remover os autocarros.
- **gerir_rotas.py**, implementação das funções que permitem ao administrador fazer a gestão (listar, adicionar e remover) das rotas.
- **mensagens.py**, implementação das funções que permitem ao administrador enviar mensagens aos clientes. (obs: apenas lidamos com as mensagens manuais neste ficheiro, ou seja, mensagens envidas pelos administradores, as mensagens automáticas são enviadas para os clientes utilizando Triggers).
- **estatisticas.py**, contem a implementação de funções que permitem tanto aos clientes como aos administradores fazerem a consulta de varias estatísticas relativas a cada um(por exemplo, o cliente consegue saber quantas viagens já realizou, fazendo também distinção de ida ou regresso de Coimbra), e ainda permitir escolher o intervalo de datas para quais elas serão calculadas.

Faltou mencionar que os clientes podem consultar as mensagens, tanto enviadas pelos admins como as automáticas (sejam elas de reserva efetuada e cancelamento efetuado), isto foi implementado no ficheiro **clientes.py** pela chamada das funções PL/SQL `open_messages()` e `consultar_mensagens_administradores` na função **cliente_menu(nif)**.

As funções mencionadas acima por sua vez utilizam a chamada de procedimentos/funções não mencionados, como:

- `filtrar_viagens` e `filtrar_reservas`, integradas na implementação das funções que permitem tanto ao cliente como ao administrador ver as viagens e no caso dos clientes as reservas, com base em filtros aplicados.
- `exists_reserva` e `exists_viagem`, verificar se existe uma reserva específica na tabela 'reservas' com base nos parâmetros fornecidos e verificar se existe uma viagem específica na tabela 'viagens' com base nos parâmetros fornecidos respetivamente.
- `stats_cliente` e `viagens_stats`, utilizadas nas estatísticas.
- `check_bus_overlap`, para verificar se o autocarro esta disponivel.

- adicionar__reserva, que adiciona uma reserva a tabela reservas, colocando-a a 'OK' ou em espera.
- atualizar__preco, como o nome diz atualiza o preço de uma viagem.

Criamos ainda Triggers para os seguintes casos:

- Atualizar os lugares disponiveis depois de uma reserva ser efetuada para uma certa viagem
- Atualizar os lugares disponiveis para uma certa viagem depois de um cancelamento
- Manter registo na tabela hsit__alteracoes as alterações de preços de viagens e por quem foram feitas.
- Ser enviada uma mensagem de confirmação de reserva ou de cancelamento (como mencionado acima)
- Atualizar o estado de uma reserva, isto é, se houver um cancelamento para uma certa viagem que estava cheia e que tem reservas em espera, essas reservas vão ocupar os lugares livres.

Por fim o ficheiro de criação das tabelas é Criar__tabela__v3 e criamos tambem uma transação para popular a nossa base de dados guardado no ficheiro trans__preencher__tabelas.

3 Screenshots ilustrativos

Nesta secção iremos mostrar alguns screenshots da nossa aplicação a correr.

3.1 Adicionar Viagens



Figure 1: Screenshot no momento de adicionar uma nova viagem pelo admin.

3.2 Listar Reservas

ID	Data Reserva	Tipo	Estado	Origem	Destino	Distancia	Data Partida	Data Chegada	Preço
3	20230515 00:00:00	Reserva	OK	Coimbra	Porto	122	20230515 10:00:00	20230515 12:00:00	50
4	20230515 00:00:00	Reserva	OK	Coimbra	Lisboa	202	20230516 14:00:00	20230516 16:00:00	70

1. Aplicar Filtros(Para ver os filtros aplicados volte aqui!)

0. Voltar

Opção:

Figure 2: Screenshot no momento de listar todas as reservas ao cliente.

3.3 Estatísticas

```
---Visualizar Estatísticas---

Selecione a data de início e de fim a visualizar as Estatísticas:
Indique a Data de Início: 2023-05-15
Indique a Data de Fim: 2023-05-16

                                Estatísticas
-----
Número de viagens                3
Preço médio                     40
Receita total                   120
Receita média por dia          120
Lugares vendidos                2
Lugares vendidos por dia       2
Melhor partida                  1
Melhor destino                  1
Número de dias                  1
```

Figure 3: Screenshot no momento em que o administrador vê as estatísticas.

3.4 Abrir mensagens enviadas pelo administrador

```
-----Mensagens-----
1.Mensagens de Administradores
2.Mensagens Automaticas de Cancelamento
3.Mensagens Automaticas de conf. Reserva
0.Voltar
Enter your choice: 1
+-----+-----+-----+-----+-----+
| id_msg | Topico | Conteudo | Remetente | lida |
+-----+-----+-----+-----+-----+
| 1 | Promoção! | Tem uma promoção no seu email! | Gonçalo Bastos | True |
+-----+-----+-----+-----+-----+
Prima enter para continuar...|
```

Figure 4: Screenshot das mensagens enviadas pelo administrador.

3.5 Reservar viagem

```
+-----+-----+-----+-----+-----+-----+-----+
| ID | Data Partida | Data Chegada | Preço | Origem | Destino | Duração | Distancia |
+-----+-----+-----+-----+-----+-----+-----+
| 4 | 20230515 10:00:00 | 20230515 12:00:00 | 50 | Coimbra | Porto | 01:20:00 | 122 |
+-----+-----+-----+-----+-----+-----+-----+
| 5 | 20230516 14:00:00 | 20230516 16:00:00 | 70 | Coimbra | Lisboa | 02:10:00 | 202 |
+-----+-----+-----+-----+-----+-----+-----+
| 6 | 20230517 18:00:00 | 20230517 20:00:00 | 60 | Porto | Coimbra | 01:20:00 | 122 |
+-----+-----+-----+-----+-----+-----+-----+
| 7 | 20230521 12:00:00 | 20230521 14:20:00 | 70 | Lisboa | Coimbra | 02:20:00 | 200 |
+-----+-----+-----+-----+-----+-----+-----+
| 8 | 20230515 20:00:00 | 20230515 22:20:00 | 23 | Lisboa | Coimbra | 02:20:00 | 200 |
+-----+-----+-----+-----+-----+-----+-----+
1. Aplicar Filtros(Para ver os filtros aplicados volte aqui!)
2. Reservar Viagem
0. Voltar
Opção: |
```

Figure 5: Screenshot no momento da reserva por um cliente onde estão a ser mostradas as viagens disponíveis.

4 Diagrama Entidade-Relacionamento (ER)

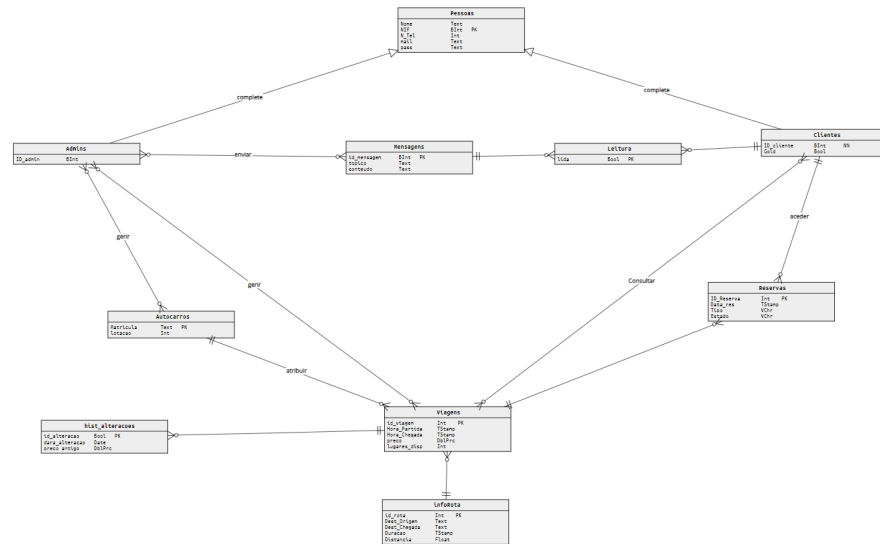


Figure 6: Diagrama ER.

Mudanças:

- Remoção da tabela estatísticas.
- Alteração de todos os id's para type SERIAL.
- Alteração do type do atributo duracao na tabela inforota
- Remoção de id_rota da tabela viagens
- Remoção da PK lida na tabela leitura
- Inserção coluna lugares_disp na tabela viagens.

5 Diagrama Físico

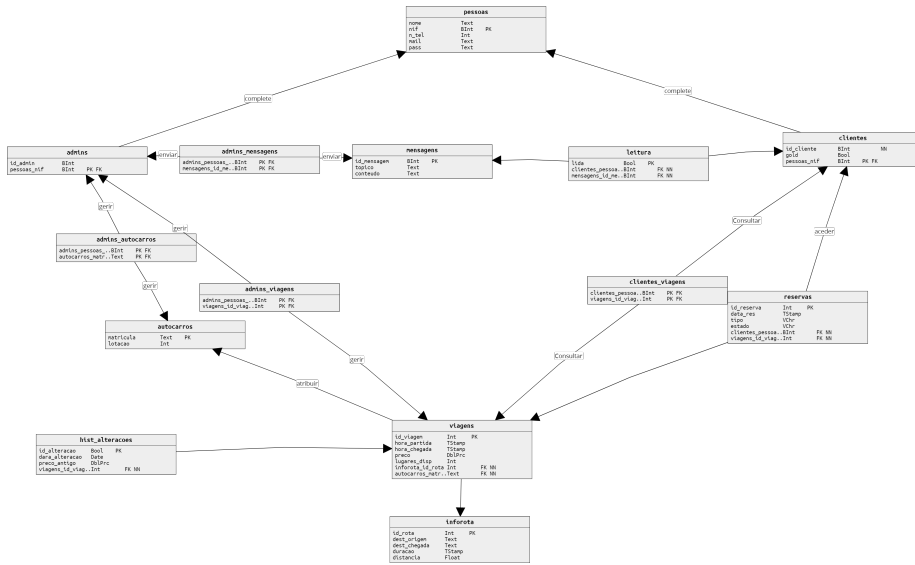


Figure 7: Diagrama Fisico.

6 Distribuição de tarefas

Gonçalo:

- Menus e controlo de fluxo do programa
- Pagina inicial, as funções de Login e Registo
- Gestão dos clientes
- Consulta e gestão das reservas
- Consulta e envio de mensagens
- Gestão de viagens

Leonardo:

- Gestão e consulta das viagens
- Apresentação das estatísticas, tanto para user como para cliente
- Gestão das rotas
- Gestão dos autocarros
- Funções de consulta e gestão das reservas

Resumindo, trabalhamos juntos nas partes mais importantes da implementação, tudo relacionado com as reservas e as viagens e dividimos tarefas no resto da implementação.

7 Tempo gasto

Leonardo	SEM 1	SEM 2	SEM 3	TOTAL
HORAS DISPONIBILIZADAS	5	14	17	36

Figure 8: Horas disponibilizadas pelo Gonçalo

Gonçalo	SEM 1	SEM 2	SEM 3	TOTAL
HORAS DISPONIBILIZADAS	5	14	17	36

Figure 9: Horas disponibilizadas pelo Leonardo.

8 Conclusão

No decorrer deste trabalho, desenvolvemos uma aplicação de base de dados para a empresa Coimbra Bus (CB), que permite a gestão de clientes, viagens, autocarros e operações relacionadas. Através desta aplicação, os clientes podem consultar viagens disponíveis, fazer reservas, gerir as suas reservas e ler mensagens dos administradores. Os administradores, por sua vez, têm acesso a funcionalidades adicionais, como gerir viagens, clientes, autocarros, enviar mensagens e ver estatísticas.

Ao longo do projeto, seguimos uma estrutura organizada, dividindo as funcionalidades em diferentes ficheiros para facilitar a manutenção e o desenvolvimento da aplicação. Implementamos funções específicas para cada tarefa, como o `login()` e o `registo()` de clientes, menus de cliente e administrador, e a gestão de viagens, reservas e mensagens.

Concluindo, este trabalho permitiu-nos aplicar os conceitos de bases de dados na prática, desenvolvendo uma aplicação completa que aborda diferentes aspetos da gestão de uma empresa de transportes. Com base nos resultados alcançados, estamos confiantes de que a aplicação desenvolvida atende aos requisitos pedidos.