**Relatório da Meta 2 do Projeto de Base de Dados**

Marta Sofia Paiva Nunes nº2017246232

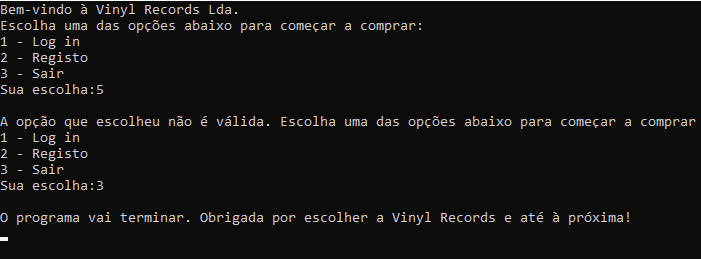
Pedro Miguel Paiva Nunes nº2016223060

**Organização final da aplicação**

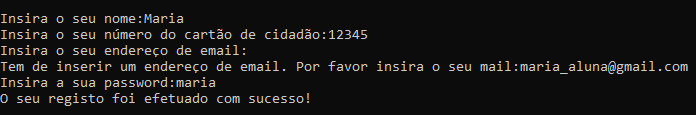
O código da nossa aplicação está dividido em dois ficheiros, um com todas as funções e um ficheiro principal, o ‘main’, que inicia o programa e chama as outras funções. Para além disso também submetemos na nossa entrega um ficheiro com o código que permite criar todas as tabelas e ligações da nossa base de dados no pgAdmin e um ficheiro com o código que permite criar o trigger.

**Aspeto geral da aplicação**

Vamos ver algumas fotos com o aspeto geral da aplicação, enquanto esta está a ser usada.

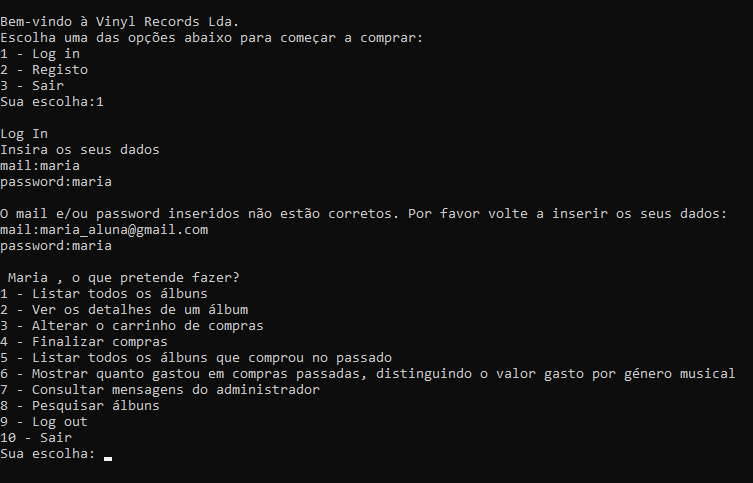


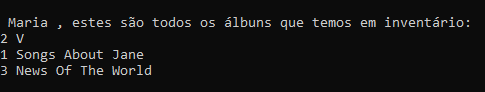
Quando o programa começa aparece-nos este menu inicial. Este menu não permite que o utilizador escolha um número diferente dos 3 números apresentados, como podemos ver. Sempre que escolhemos sair do programa aparece a mensagem a avisar que o programa vai terminar e uns segundos depois o programa fecha.



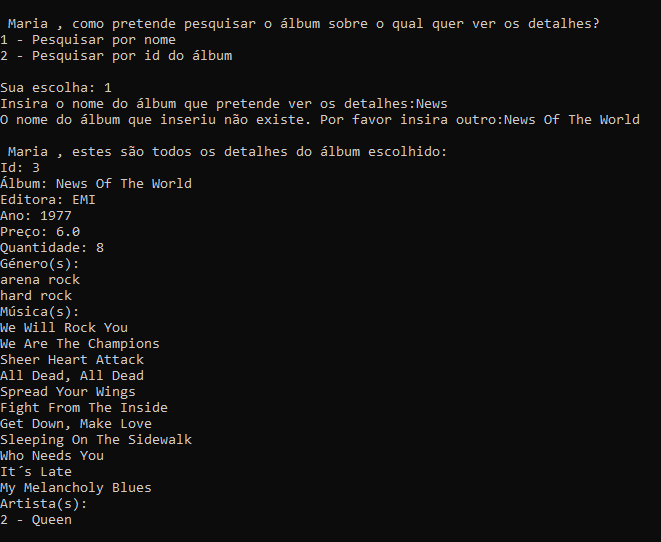
Ao escolhermos a opção de registo o programa vai pedindo os dados necessários para o registo de um novo cliente. Como podemos ver, não é permitido o utilizador inserir strings vazias. Quando o registo é efetuado o cliente é avisado que o seu registo foi efetuado com sucesso.

Quando escolhemos a opção de Log In, o programa pede as credenciais necessárias ao Log In. Se a password ou o endereço do mail estiverem incorretos, o log in não é efetuado e é pedido que insiram novamente as credenciais. Se as credenciais estiverem corretas, o programa vai para o menu inicial do cliente ou do administrador, dependendo de quem fez o Log In. De seguida iremos ver as opções do cliente.



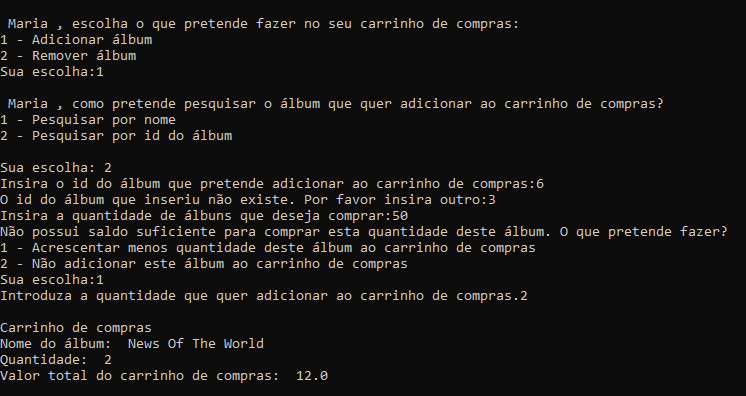


No menu inicial do cliente este pode escolher entre as várias opções disponíveis. Na imagem anterior podemos ver o caso em que o cliente escolhe ver todos os álbuns existentes no inventário, que neste caso são os 3 álbuns apresentados em cima.

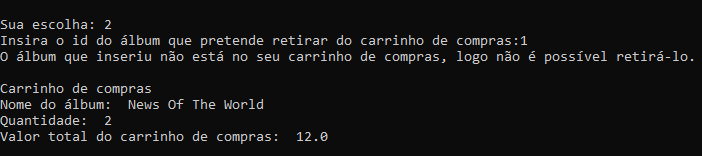


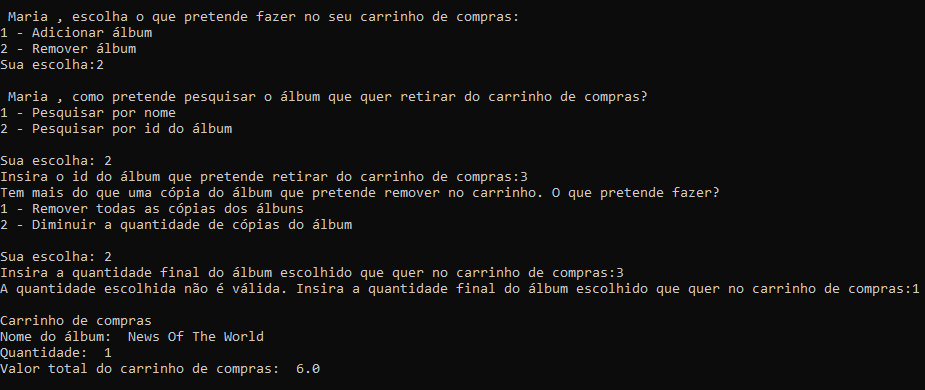
Podemos agora ver o caso em que o cliente escolhe ver os detalhes de um álbum. Esta opção permite que o cliente escolha como pretende pesquisar o álbum: por nome ou por id. Após inserir um nome ou um id válido, são apresentados todos os detalhes de um álbum, como podemos ver em cima.

Vejamos de seguida o que acontece quando o utilizador escolhe alterar o seu carrinho de compras. Ao escolher esta opção é dado a escolher se o cliente pretende adicionar ou remover um álbum do carrinho, e como pretende pesquisar este álbum. Se o cliente não inserir um id ou inserir um id que não exista é pedido outro, como podemos ver na imagem (o mesmo se aplica ao nome de álbum). Se o álbum que o cliente pretende inserir no seu carrinho não tiver stock o cliente é avisado de que não é possível adicionar este álbum ao carrinho e volta ao menu inicial do cliente. Por fim, é pedido ao cliente para adicionar a quantidade do álbum que pretende adicionar ao carrinho. Se não houver stock suficiente ou o cliente não tiver saldo suficiente são dadas as opções de adicionar menos quantidade do álbum em questão ou não adicionar de todo o álbum ao carrinho. Após o cliente inserir uma quantidade de álbuns que existe no stock e que tem saldo para comprar é mostrado os detalhes do seu carrinho de compras, que incluem o nome do álbum, a quantidade no carrinho e o valor total do carrinho.

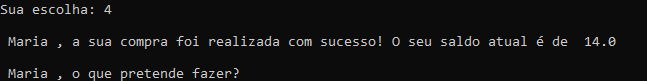


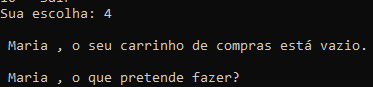
Se o cliente pretender retirar um álbum poderá novamente escolher como deseja pesquisar o álbum que pretende retirar, se por nome ou por id, sendo que o álbum inserido tem de ser válido, tem de existir no stock total. Ao escolher o álbum que pretende retirar do carrinho, existem algumas coisas que podem acontecer. Se o carrinho estiver vazio não será possível pesquisar um álbum para retirar do carrinho, uma vez que não existe nenhum álbum. Se o cliente tentar retirar um álbum que existe, mas que não está no seu carrinho, recebe uma mensagem com esta informação e, depois de ver o seu carrinho, volta para o menu inicial, como vemos na figura seguinte. Se o cliente tentar remover um álbum que existe no seu carrinho este é removido se a sua quantidade no carrinho for igual a 1. Caso a sua quantidade seja maior que um, o cliente pode escolher se retira o álbum ou se só diminui a quantidade deste. No fim da operação o carrinho de compras é sempre mostrado.



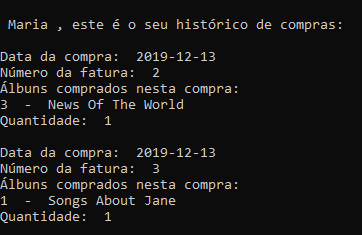


Ao escolhermos a opção finalizar compras, a compra é realizada e aparece uma mensagem sobre como correu a operação e o saldo final, como vemos nas seguintes imagens. Após finalizar uma compra o carrinho de compras é esvaziado. Se tentarmos realizar alguma compra com o carrinho de compras vazio aparece a mensagem que vemos nas imagens seguintes. Se por alguma razão a compra não for efetuada com sucesso, o programa não vai abaixo, mostra sim qual foi o erro, uma vez que a transação está a ser efetuada neste ponto. O trigger também está a ser feito nesta parte do código.

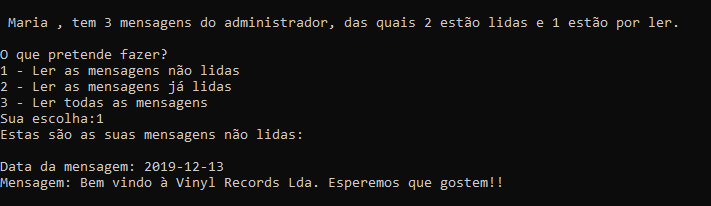




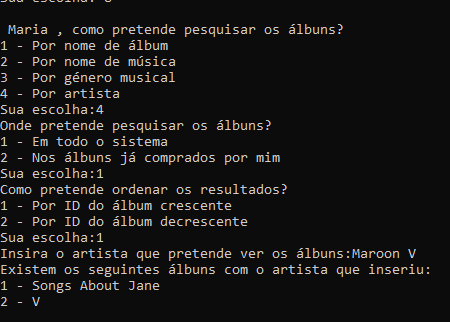
Quando pretendemos visualizar o histórico de compras poderemos ver algo parecido com a imagem seguinte. Se não tivermos realizado nenhuma compra, receberemos essa mensagem. Caso já tivermos efetuado compras veremos os seus pormenores, como a data da compra, o número da fatura e os álbuns comprados e a quantidade de cada álbum que foi comprada. Se houver mais do que um álbum diferente comprado na mesma compra estes vão aparecer todos incluídos na informação da mesma compra, não em compras separadas.



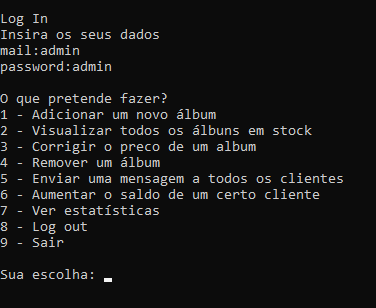
Na figura seguinte podemos ver o que acontece quando pretendemos ver as mensagens. Inicialmente é mostrado quantas mensagens existem, especificando quantas estão lidas e quantas estão por ler. Se não existir nenhuma, aparece uma mensagem a informar e vamos para o menu inicial do cliente. De seguida, o cliente pode escolher se quer ler as mensagens que ainda não foram lidas, as que já foram lidas ou todas. As mensagens inseridas antes da data de registo do cliente não serão apresentadas a esse cliente.



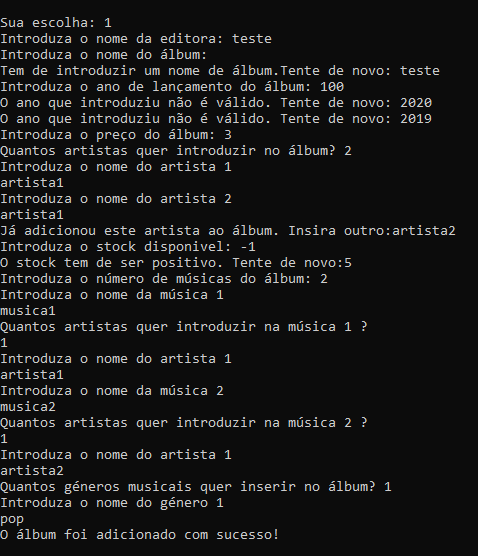
Por último, o menu do cliente permite pesquisar álbuns de diferentes maneiras. Quando escolhemos esta opção podemos escolher entre as várias maneiras de pesquisar um álbum e se pretende pesquisar o álbum em todo o sistema ou nos álbuns comprados pelo cliente. Por último, o cliente pode ainda escolher como pretende ordenar os resultados. No exemplo em baixo temos o caso de pesquisar um álbum por artista em todo o sistema, ordenando os resultados por ID do álbum crescente.



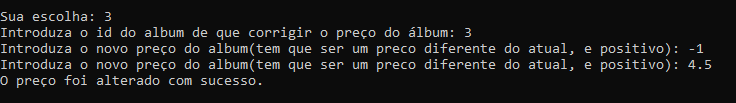
Vejamos agora a parte do programa destinada ao administrador. O Log In do administrador funciona da mesma maneira que o Log In do cliente, sendo que é no código do programa que está a distinção de que menu é apresentado. Na figura seguinte podemos ver o menu do administrador.



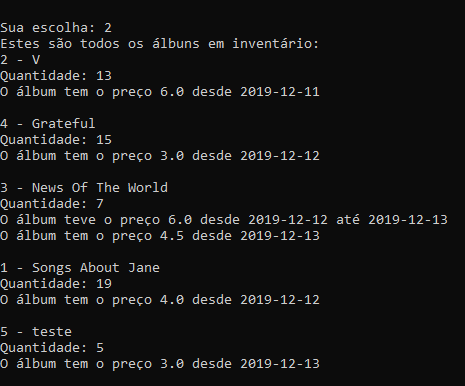
Na figura seguinte podemos ver como se insere um álbum. A partir deste exemplo vemos que não podemos inserir strings vazias e que se quisermos inserir mais do que um artista podemos fazê-lo, mas não podemos inserir o mesmo artista duas vezes no mesmo álbum. Para além disso temos de inserir um ano válido, que não pode ser um ano do futuro. Também podemos inserir mais do que um género, mas se quisermos inserir mais do que um não poderão ter todos o mesmo nome, os nomes têm de ser distintos. Quando o álbum foi inserido o administrador recebe uma mensagem informativa.



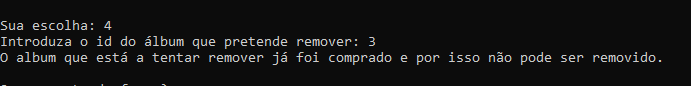
Vejamos agora o que acontece quando queremos corrigir o preço de um álbum. Se não existirem álbuns na base de dados, recebemos uma mensagem informativa e voltamos ao menu do administrador. Caso contrário, é pedido o ID do álbum e o novo preço, que tem de ser diferente do anterior e positivo. Quando o preço é alterado o administrador recebe uma mensagem informativa.

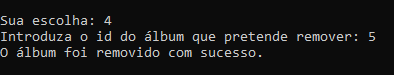


O administrador pode, tal como o cliente, visualizar todos os álbuns em stock. No entanto, o administrador pode ver mais informação, como o histórico de correções de preço. Como podemos ver na figura, para cada álbum aparecem as alterações realizadas. (Nota: nesta listagem de álbuns aparecem mais álbuns que na listagem do cliente pois os álbuns foram adicionados enquanto fazíamos o relatório, para testarmos as funções)

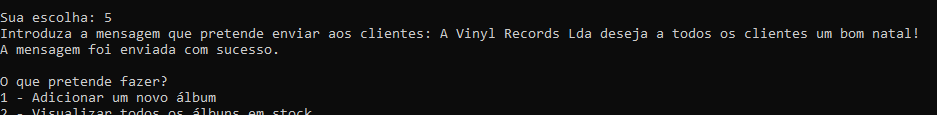


Vejamos agora a opção de remover um álbum. Temos dois casos expostos nas seguintes imagens. No caso de o álbum já ter sido comprado não é possível remover o álbum e o administrador é informado disso. No caso de o álbum ainda não ter sido comprado, este é removido e é enviada uma mensagem informativa. Ao removermos um álbum não estamos a remover os seus géneros e os seus artistas, uma vez que estes podem ter outros álbuns.

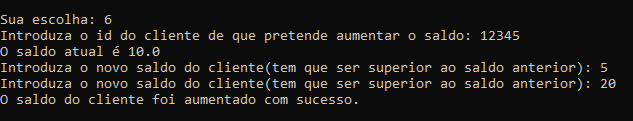




Para enviar uma mensagem basta o administrador escolher a opção de enviar mensagem e escrever o texto. O programa não deixa enviar uma mensagem vazia.

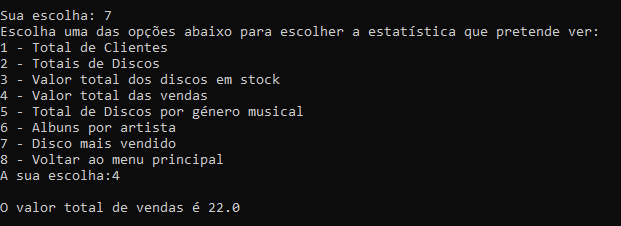


O administrador pode aumentar o saldo de um cliente. Para isso basta inserir o ID de um cliente e inserir o novo saldo do cliente, que tem de ser superior ao saldo anterior. Quando o saldo for aumentado o administrador recebe uma mensagem informativa.



Por fim, o administrador pode visualizar várias estatísticas. As 2 estatísticas extra que inserimos foram mostrar os álbuns que existem por artista e qual o disco mais vendido. Na seguinte imagem podemos ver o exemplo do valor total de vendas.

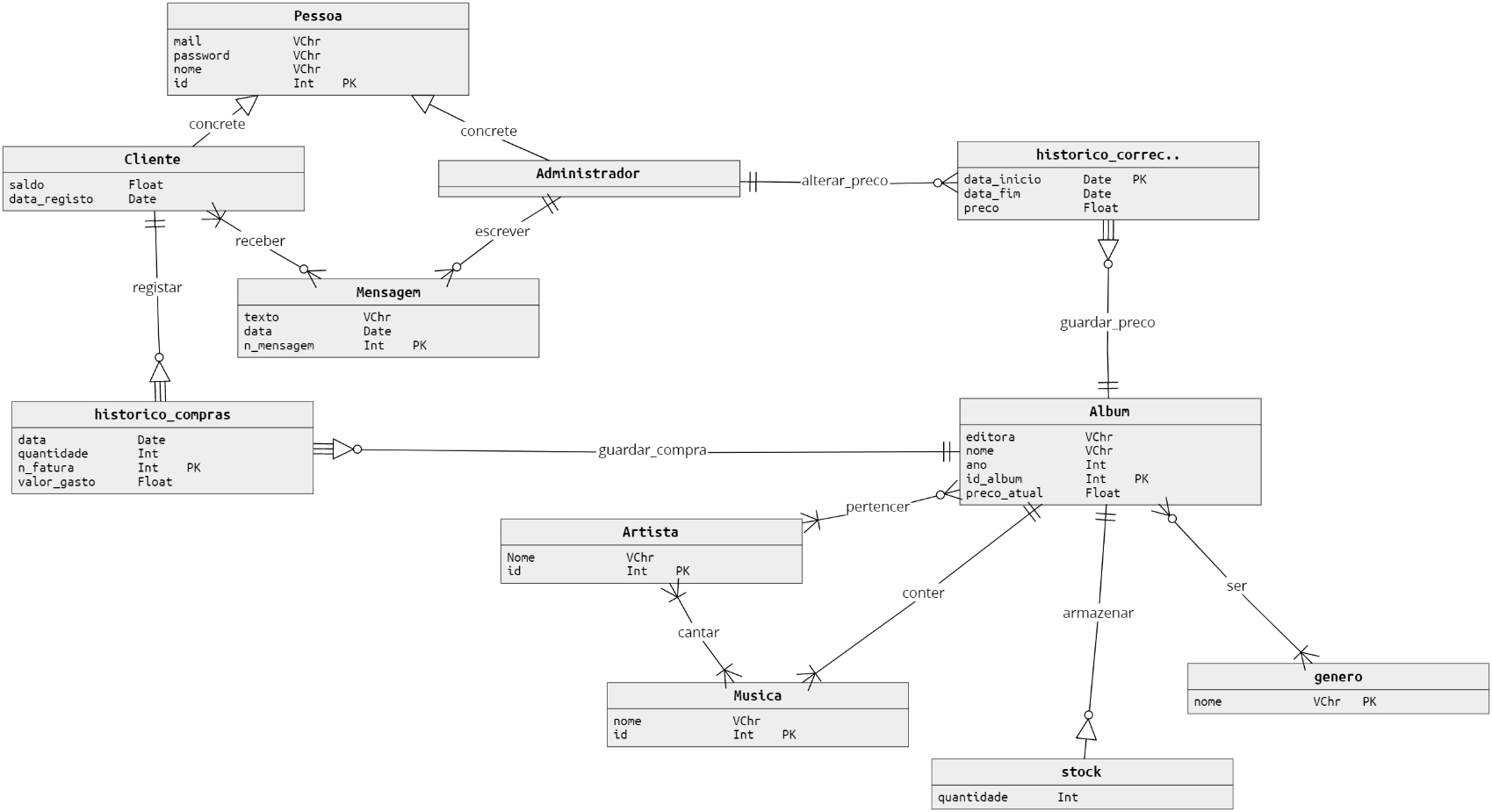
Para além destas opções, tanto o cliente como o administrador têm opção de fazer Log Out, sendo que quando o cliente o faz o seu carrinho de compras é apagado.



**Diagrama conceptual**

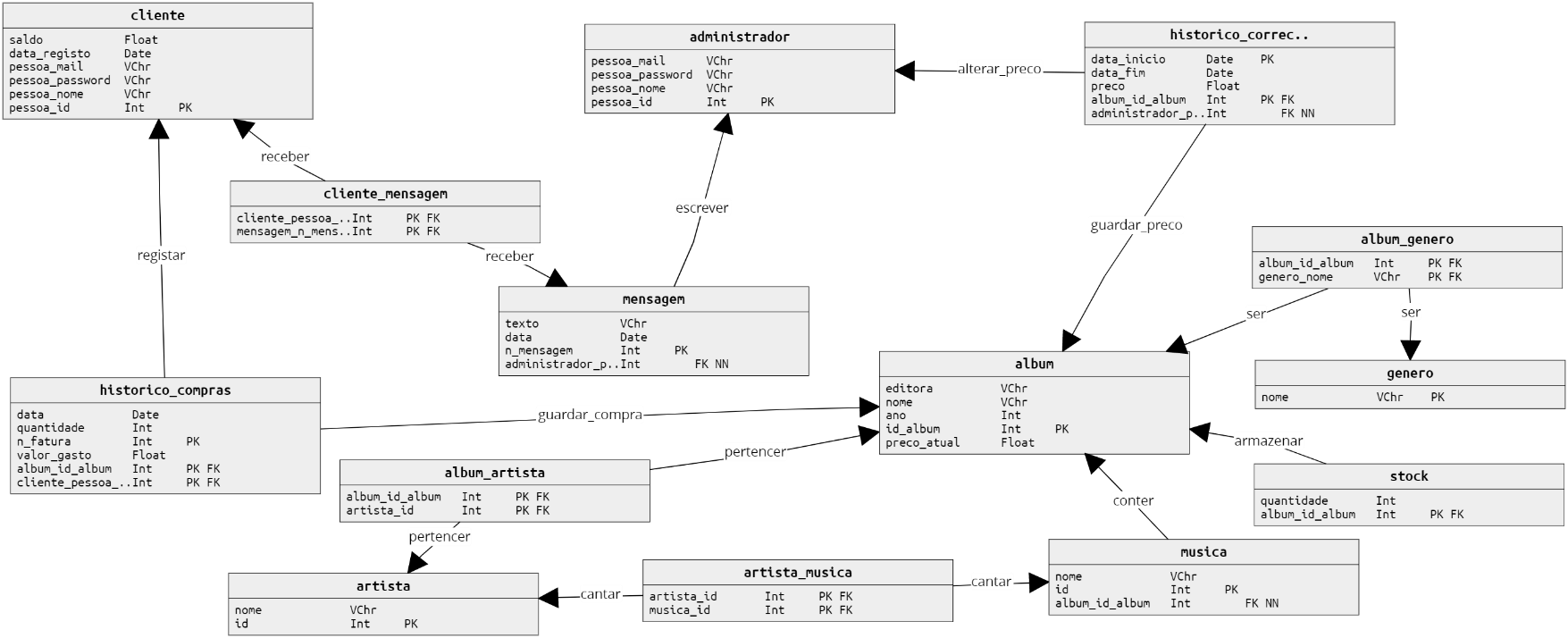
Após a defesa da meta 1 foram-nos apontados alguns erros no diagrama conceptual. Para além disso, durante a realização do projeto também detetámos algumas gralhas. Assim, alterámos o diagrama para o mostrado na foto que segue a seguir. As alterações feitas foram:

* Entidade cliente – nesta identidade adicionámos a data de registo do cliente para que o cliente só possa ver as mensagens que foram enviadas após a sua data de registo. Esta data é adicionada automaticamente.
* Entidade mensagem – após a defesa da meta 1 percebemos que a identidade verificação\_mensagem que tínhamos não fazia sentido, pelo que foi apagada. Também alterámos o tipo de variáveis dos vários campos e acrescentámos um número de mensagem que passou a ser a nossa chave primária.
* Entidade histórico\_compras – nesta entidade as únicas alterações que fizemos foi acrescentar campos na entidade, por percebermos que estes nos faziam falta. Os campos que acrescentámos foi o número da fatura, que é a nossa chave primária, a quantidade que o cliente comprou do álbum e valor gasto pelo cliente na compra desses álbuns.
* Ligação alterar\_preco – na ligação entre o administrador e o histórico de correções alterámos a cardinalidade do administrador de 1-n para 1-1.
* Entidade álbum – esta entidade foi a que sofreu mais alterações, uma vez que eliminámos alguns dos atributos e acrescentámos entidades dos atributos que eliminámos. Os atributos eliminados foram o artista e o género. Para além disso o preco\_atual foi alterado para float.
* Entidade género – a entidade género foi uma das duas entidades que foram acrescentadas. O único atributo desta entidade é o nome, que é a chave primária.
* Ligação ser – ao acrescentarmos a entidade género tivemos de acrescentar uma ligação entre o género e o álbum. Esta ligação tem uma cardinalidade do álbum 0-n e uma cardinalidade do género 1-n.
* Entidade artista – esta entidade foi a outra entidade acrescentada. Os seus atributos são o id do artista, que é a chave primária e o nome do artista.
* Ligação pertencer – ao criarmos a entidade artista tivemos que a ligar com a entidade álbum, com esta ligação. A cardinalidade do álbum é 0-n e a cardinalidade do artista é 1-n.
* Ligação cantar – com a criação da entidade artista também criamos a ligação cantar, que liga o artista com a música. Tanto a cardinalidade no lado do artista como no lado da música são 1-n.
* Entidade música – nesta entidade acrescentámos o id da música, que é a chave primária.
* Ligação conter – a ligação entre o álbum e a música foi alterada, sendo que a cardinalidade do lado da música é 1-n.



**Diagrama físico**

Como consequência da alteração do diagrama conceptual, o diagrama foi também alterado em relação à meta I deste trabalho. As alterações foram as seguintes:

* Tabelas administrador e historico\_correcoes: Foi retirada a tabela que ligava às duas tabelas passando a estar relacionadas diretamente
* Tabelas cliente e mensagem: Em vez de ter uma tabela de verificação de mensagem tanto para o cliente como para a mensagem, agora existe apenas a tabela cliente\_mensagem que permite a verificar se as mensagens do cliente, que são enviadas pelo administrador, já foram lidas
* A ligação entre as tabelas cliente e historico\_compras mantiveram-se
* A antiga tabela album\_stock foi substituída pela tabela album
* De modo a ser possível implementar todas as funcionalidades do projeto foram acrescentadas as tabelas album\_artista, artista, artista\_musica, musica, álbum\_genero, genero e stock
* A tabela album passa a estar ligada à tabela album\_artista
* A tabela album\_artista está ligada à tabela artista
* A tabela artista está ligada à tabela artista\_musica
* A tabela artista\_musica está ligada à tabela musica
* A tabela album, à semelhança da antiga tabela álbum\_stock, está ligada à tabela musica
* A tabela album está ainda ligada à tabela album\_genero
* A tabela album\_genero está ligada à tabela genero
* A tabela album está ligada à tabela stock

**Distribuição final de tarefas por elemento**

Tal como o previsto, um dos elementos do grupo ficou encarregue de fazer as funções relacionadas com o cliente e o outro elemento ficou responsável pelas funções relacionadas com o administrador. Antes desta distribuição foram feitos o Log In e o Registo por um dos elementos, de maneira a ser um início para o projeto. Depois de acabarmos o código, ambos os elementos o testaram e corrigiram os eventuais erros que surgiram.

**Tempo gasto com a disciplina**

Cada aluno gastou, por semana, as horas das aulas teóricas e práticas e cerca de 1/2 horas adicionais. Desde que começámos o projeto temos dispendido uma média de cerca de 2 horas por dia para fazer o projeto.