



deti

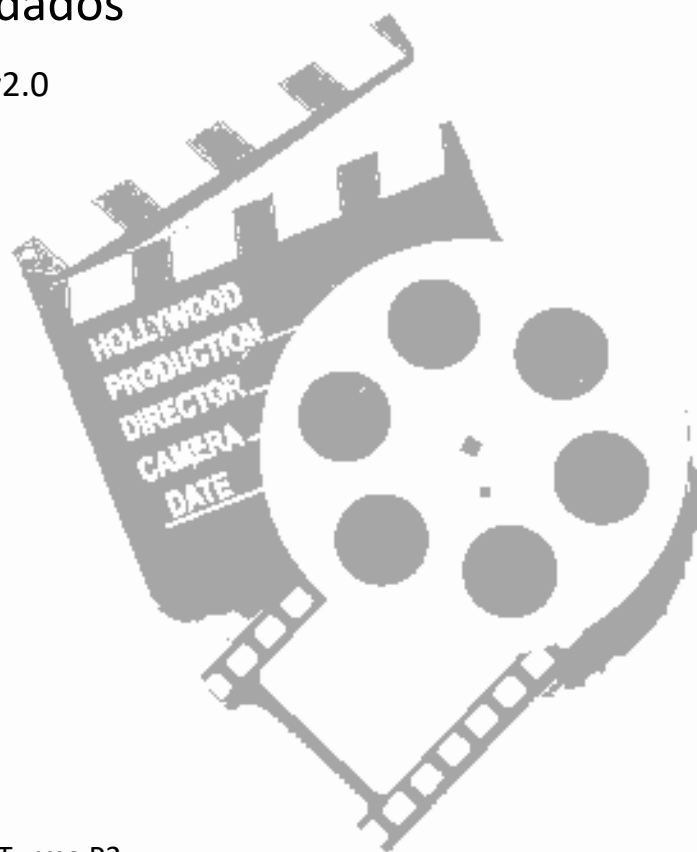
universidade de aveiro
departamento de electrónica,
telecomunicações e informática

VideoClube

LECI - Base de dados

Relatório Final v2.0

23/06/2022



Turma P2

Grupo 7:

Frederico Viera, Nmec:98518

Tiago Coelho, Nmec:98385

Conteúdo

Introdução	3
Análise de requisitos	4
Problema.....	4
Solução:	4
Requisitos:	4
Diagrama Entidade-Relacionamento.....	5
Esquema Relacional.....	6
Backlog	7
SQL DDL	9
Setup.....	9
Drop.....	9
Inserts	9
Test Querys	9
SQL DML	10
Indexes.....	10
Stored Procedures.....	10
Triggers	11
UDF's.....	12
Teste	13
Para Testar a aplicação - Como Cliente	13
Para Testar a aplicação - Como Admin	13

Introdução

No âmbito da disciplina de Base de dados da Licenciatura em Engenharia de Computadores e Informática, foi nos requisitado a planeamento e desenvolvimento de um projeto de um sistema de base de dados.

Para o planeamento usamos a ferramenta de gestão de backlog “Pivotal Tracker”, por forma a fazer um planeamento ágil, tendo como entradas do backlog user stories definidas e planeadas em grupo. Cada user storie possui também uma breve descrição com os requisitos a implementar e tarefas a fazer.

Para o desenvolvimento, escolhemos como tema de trabalho a transformação digital de um videoclube, escolhendo um sistema que considerámos ter problemas (videoclubes antigos físicos), transformando digitalmente numa plataforma de streaming de aluguer de filmes e séries. A construção foi realizada em Microsoft SQL Server Management Studio para desenvolver a base de dados em SQL e Visual Studio 2019 para desenvolver a interface em Windows Forms e Visual Basic.

Análise de requisitos

Problema:

O objetivo seria assim resolver os problemas relacionados com os videoclubes físicos, como por exemplo, nestes estabelecimentos só se continha apenas alguns exemplares dos filmes, podendo esgotar facilmente, também havia outro problema quando existia atrasos na entrega do filme, e também podia o mesmo regressar com algum dano de tal forma que não funcionaria mais.

Solução:

Ao transformar, pretendemos corrigir todos esses problemas indicados, assim como adicionar mais funcionalidades que possam ser uteis ao utilizador. Outro benefício é em comparação com as plataformas de streaming com mensalidades (Netflix, HBO Max, ...), visto que para as mesmas necessitamos de pagar o valor de uma mensalidade, e no caso de pretendemos ver apenas um filme/série temos de pagar a totalidade da mensalidade, algo que não aconteceria na plataforma de aluguer.

Requisitos:

Na nossa plataforma vão existir duas entidades, Utilizador e Admin, o primeiro é considerado um cliente normal da plataforma e o segundo é considerado um dos responsáveis por gerir a plataforma.

Estas são as funcionalidades presentes na plataforma divididas em Utilizador e Admin:

Utilizador pode:

- Criar Conta e ligar-se na mesma;
- Procurar um filme/série e alugar o mesmo;
- Pesquisar por um filme/série;
- Ordenar por preço, ano e título;
- Filtrar por categorias e/ou por filmes ou séries;
- Usar/ganhar pontos no aluguer;
- Usar cupão do mês atual e ver promoções;
- Ver dados de utilizador e histórico de aluguer e cupões;

Admin pode:

- Ligar-se na conta;
- Adicionar conteúdos;
- Editar conteúdos;
- Adicionar admins;
- Adicionar promoções/cupões;

Diagrama Entidade-Relacionamento

Um User é caracterizado pelo Email, Nome, Contribuinte, Data Nascimento e Pontos.

Um Cupão é caracterizado pelo ID, data início, data fim, pontos. Pode ser utilizado pelo User e é gerido pelo admin.

Um Admin é caracterizado pelo Email, Nome, Password. Gere o cupão, promoção e conteúdos

Uma Promoção é caracterizado pelo ID, Nome, Desconto, data início, data fim está associado ao Conteúdo e é gerido por um Admin.

Um Conteudo_taxa é caracterizado por a Taxalva.

Um Tipo_conteudo é uma Serie ou Filme e é caracterizado pela Categoria.

Um Filme e Série são tipos de conteúdo que se pode alugar. No caso das séries temos o número de episódios.

Um Conteúdo é caracterizado pelo ID, Ano, Preço Idade, Duração e Título. É gerido pelo Admin, tem um tipo associado ao Tipo_conteúdo e tem uma promoção.

Um aluguer é caracterizado pelo ID, Data Início, Data Fim, Pagamento, Contribuinte, Pontos Usados e Pontos Gastos. Está associado com o conteúdo podendo ter N.

Após a primeira entrega houve diversas alterações para corrigir erros e colocar as devidas entidades e atributos referentes ao novo Sistema de Pontos com vendas e cupões e Promoções de conteúdo.

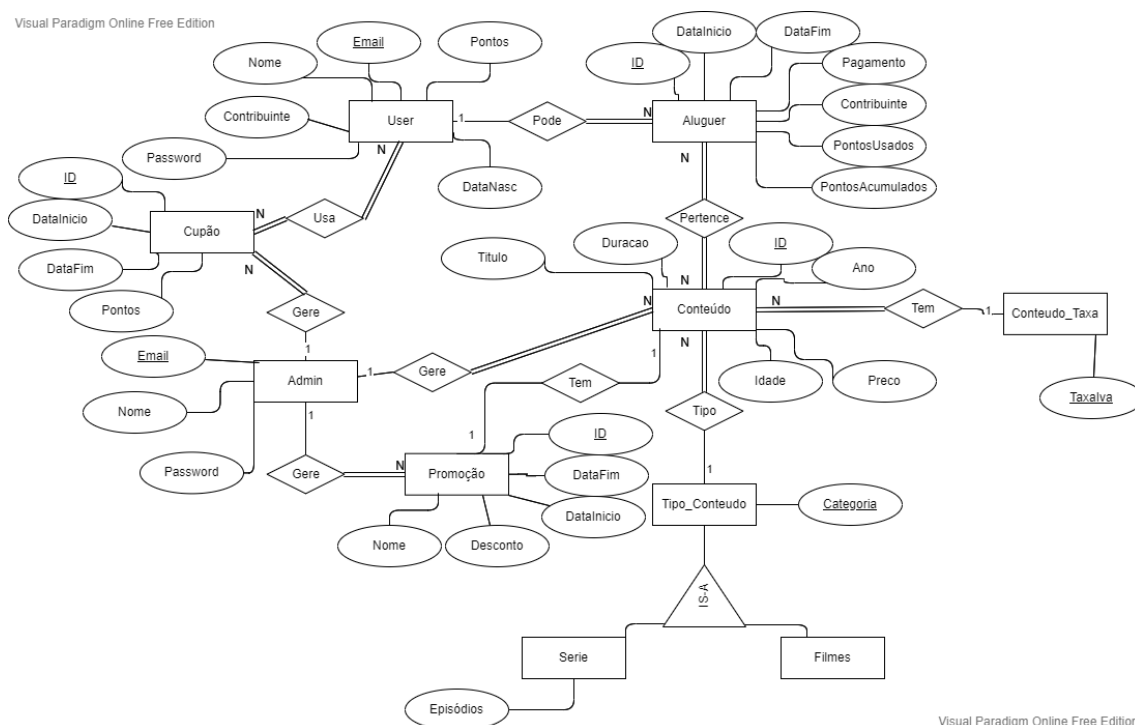


Figura 1 / Diagrama Entidade-Relacionamento

Esquema Relacional

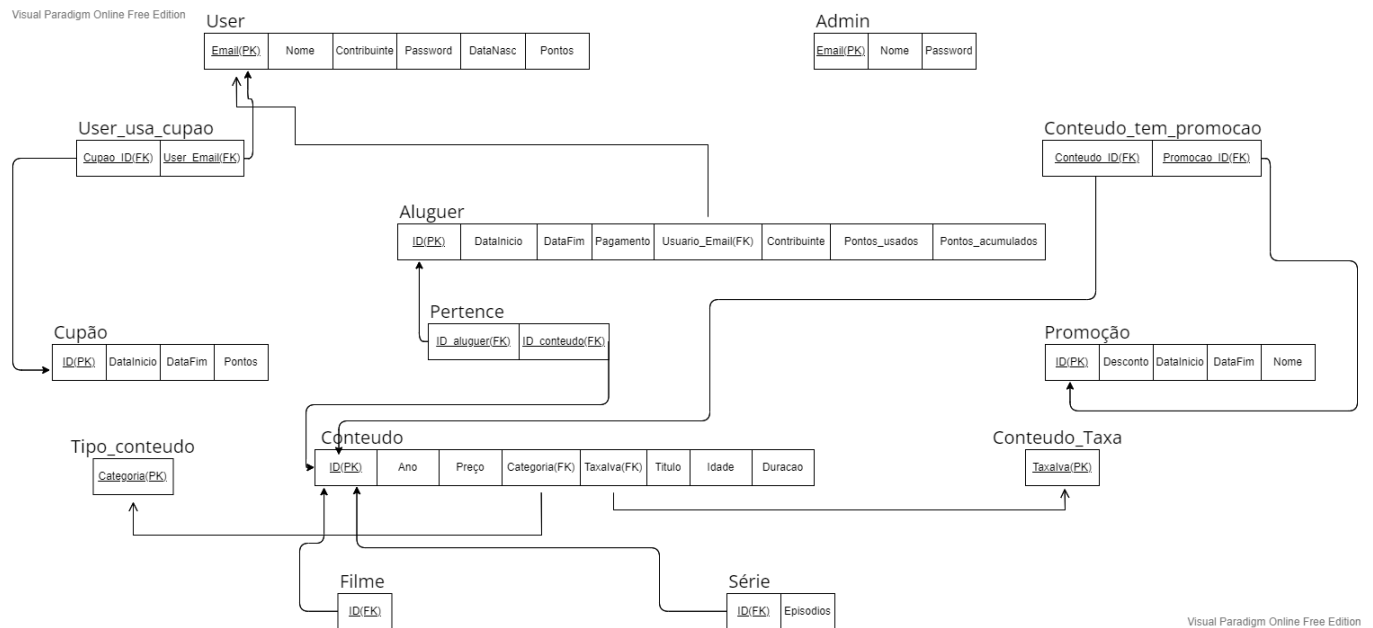


Figura 2 | Esquema Relacional

Após a primeira entrega houve diversas alterações para corrigir erros e colocar as devidas tabelas referentes ao novo Sistema de Pontos com vendas e cupões e Promoções de conteúdo.

Backlog

Como anteriormente referido foi usado a ferramenta online que serviu de suporte ao planeamento ágil. As entradas do backlog são user stories definidas e planeadas em grupo. Foi lhes dada uma estimativa de esforço e prioridade. Cada user storie possui também uma descrição da user storie, com os requisitos a implementar e um conjunto de tarefas a fazer.

Em baixo poderá encontrar prints das entradas do backlog.

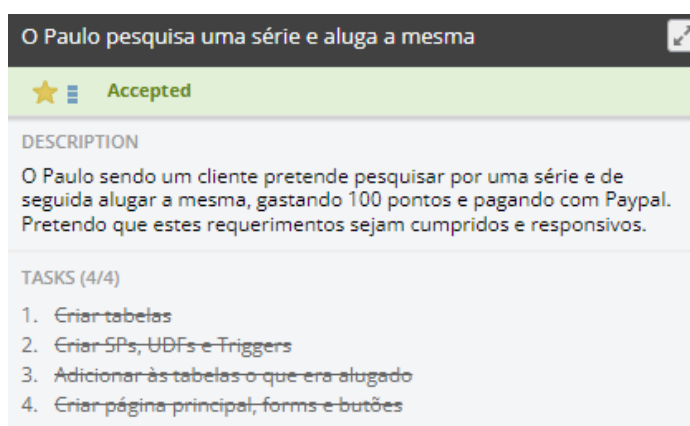


Figura 3| User Storie Utilizador

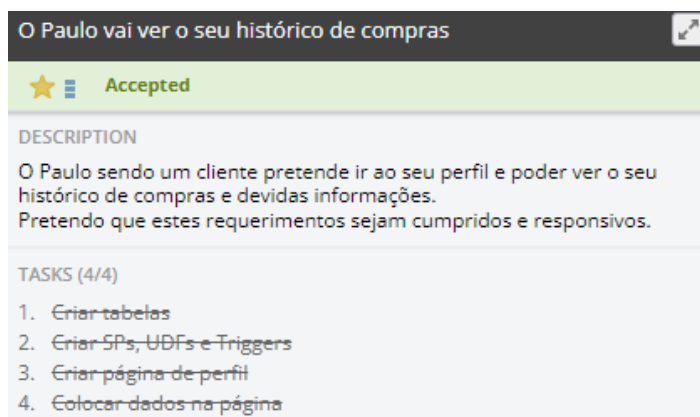


Figura 4| User Storie Utilizador

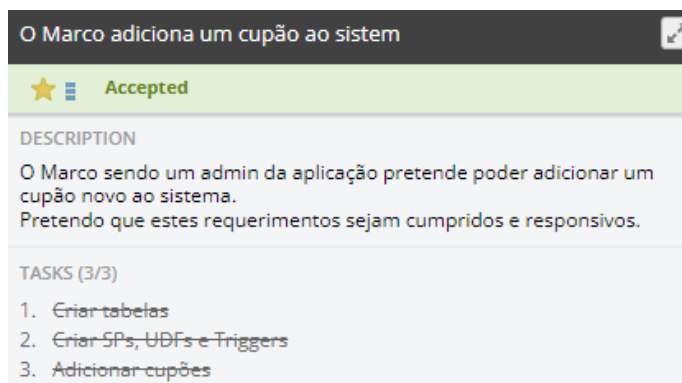


Figura 5| User Storie Utilizador

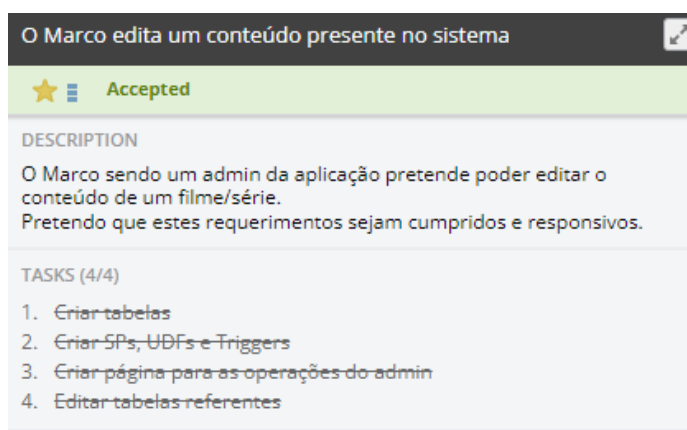


Figura 6| User Storie Admin

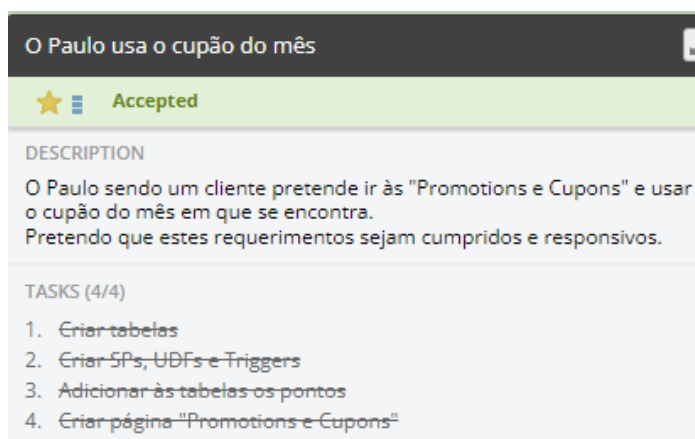


Figura 7| User Storie Admin

SQL DDL

Setup

O ficheiro `setup_VideoClub.sql` contém o código sql que permite a criação das tabelas da aplicação, descritas anteriormente no diagrama relacional.

Drop

O ficheiro `drops_VideoClub.sql` contém código sql que permite a eliminação das tabelas do projeto, descritas anteriormente no diagrama relacional.

Inserts

O ficheiro `inserts_VideoClub.sql` contém código sql que permite a inserção de dados nas tabelas, descritas anteriormente no diagrama relacional.

Test Querys

O ficheiro `StartTests_VideoClub.sql` contém código sql que foi utilizado no projeto inicialmente para testes.

SQL DML

Indexes

O ficheiro idx_VideoClub.sql contém código sql que cria todos os Indexes.

Para o nosso projeto foi utilizado apenas um index relaciona com a tabela Conteudo para as colunas Preco, Categoria, Taxalva, Título e Idade.

Stored Procedures

O ficheiro sps_VideoClub.sql contém código sql que cria todos os Stored Procedures.

Os procedures do projeto foram utilizados com o intuito de criar uma espécie de camada de abstração entre a Base de Dados e as operações selecionadas através da interface de modo a não pôr em risco a integridade dos dados.

Stored Procedure	Descrição
addConteudo	Adiciona um novo Conteúdo à base de dados, recebe como parâmetros o email do admin que está a adicionar e os dados necessário para a tabela Conteudos e o responseMessage como output para dar algum feedback do resultado. Foi utilizado o NOCOUNT ON.
addCoupon	Adiciona um novo Cupao à base de dados, recebe os parâmetros necessários para a tabela Cupao e o responseMessage como output para dar algum feedback do resultado. Foi utilizado o NOCOUNT ON.
addPromocao	Adiciona uma nova Promoção à base de dados, recebe como parâmetros o email do admin que está a adicionar e os dados necessário para a tabela Promoção e o responseMessage como output para dar algum feedback do resultado. Foi utilizado o NOCOUNT ON.
AlugarConteudo	Adiciona uma nova linha na tabela Pertence à base de dados, que contém id_aluguer e id_conteudo recebe como parâmetros o id_aluguer e id_conteudo, tem responseMessage como output para dar algum feedback do resultado. Foi utilizado o NOCOUNT ON.
ChangeConteudo	Faz alterações num Conteúdo existente na base de dados, recebe como parâmetros o email do admin que está a alterar e os dados necessário para a tabela Conteudos e o responseMessage como output para dar algum feedback do resultado. Foi utilizado o Update para alterar os dados, o NOCOUNT ON e TRANSACTION.
Conteudo_filter	Faz a pesquisa/filtragem de um conteudo através do titulo do mesmo ou através da categoria e é possível ordenar os conteudo por titulo,categoria,preço ou ano. Recebe como parâmetros o titulo,categoria e orderby apenas parametros necessários para a realizar da pesquisa. Foi utilizado o NOCOUNT ON.
getlastaluguer	Vai buscar o id do último aluguer utilizado e return esse id numa responseMessage.
login	Faz o login na aplicação e dá acesso aos utilizadores e também é bastante importante para distinguir se quem está a fazer o login é um admin ou um utilizador normal. Recebe como parâmetros o username e password. E devolve uma mensagem, um type que é

	do tipo Bit e o email já introduzido se o login tiver sido bem-sucedido. Foi utilizado o NOCOUNT ON.
novoAluguer	Adiciona um novo Aluguer à base de dados, recebe como parâmetros necessários para a tabela aluguer e o responseMessage como output para dar algum feedback do resultado. Foi utilizado o NOCOUNT ON e TRANSACTION.
Register_Admin	Adiciona um novo admin à base de dados, recebe como parâmetros necessários para a tabela admin e o responseMessage como output para dar algum feedback do resultado. Foi utilizado o NOCOUNT ON e TRANSACTION.
RegisterUsuario	Adiciona um novo usuario à base de dados, recebe como parâmetros necessários para a tabela usuario e o responseMessage como output para dar algum feedback do resultado. Foi utilizado o NOCOUNT ON e TRANSACTION.
Usuario_usa_cupao	Adiciona uma nova linha na tabela usuario_usa_cupao à base de dados, que contém cupao_id e usuario_email recebe como parâmetros o cupao_id e usuario_email, tem responseMessage como output para dar algum feedback do resultado. Foi utilizado o NOCOUNT ON.
updateUsuario	Altera os pontos um usuario já existente na base de dados, recebe como parâmetros o email do usuario e o pontos para adicionar assim como responseMsg como output para dar algum feedback do resultado. Foi utilizado o NOCOUNT ON e TRANSACTION.
deleteadmin	Elimina admins existentes na base de dados, recebe como parâmetros o email do admin que irá ser eliminado assim como responseMessage como output para dar algum feedback do resultado. Foi utilizado TRANSACTION.
updateadmin	Altera os dados de um admin já existente na base de dados, recebe como parâmetros o email, nome e pass do admin para eliminar assim como responseMsg como output para dar algum feedback do resultado. Foi utilizado o NOCOUNT ON e TRANSACTION.

Tabela 1 | Stored Procedures

Notes: NOCOUNT ON -> O nocount on faz com que não seja contado qualquer linha assim evita mostrar as linhas afetadas.

TRANSACTION -> Faz com que todas as operações sejam realizadas ou nenhuma será. Bastante util para manter a integridade dos dados.

Triggers

O ficheiro triggers_VideoClub.sql contém código sql que cria todos os triggers.

Foi apenas desenvolvido um trigger do tipo AFTER na nossa aplicação foi o “usaCupaoTrigger” associado à tabela [usuario_usa_cupao] faz com que, sempre que um utilizador resgate um cupão dos que estão listados na loja, o utilizador ganhe os pontos desse cupão fazendo a soma dos pontos do cupão com os pontos pessoais do utilizador.

UDF's

O ficheiro udfs_VideoClub.sql contém código sql que cria todas as UDF's.

Os UDF's vão ser utilizadas apenas para devolver tabelas que vão ser necessárias.

UDF	Descrição
getadminInfo	Devolve todos os admins da tabela admin
getallConteudo	Devolve todos os conteudos da tabela Conteudo
getallConteudo_filme	Devolve apenas os filmes dos Conteudos
getallConteudo_serie	Devolve apenas as Series dos Conteudos
getConteudobytitulo	Devolve os Conteudos baseados no titulo
getCupao	Devolve apenas os cupões que tenham a data válida
getPromocoas	Devolve todas as promoções
getUsuarioCoupao	Faz com que seja possível cada utilizador ver o seu histórico de cupões utilizados.
getUsuarioInfo	Devolve a informação de um determinado utilizador
historico_aluguer	Faz com que seja possível cada utilizador ver o seu histórico de alugueres.

Tabela 2 | UDF's

Teste

Para Testar a aplicação - Como Cliente

Fazer login com um user já existente (por exemplo user:joaopedro@gmail.com pass:jpedro987) ou então efetuar um registo próprio depois de entrar na app é possível fazer as pesquisas dos conteúdos listados, se clicar em algum dos conteúdos vai abrir um form para alugar o conteúdo vai haver a possibilidade de usar pontos e o preço do filme vai alterando cada euro pago é igual a 100 pontos acumulados. É possível ir ao profile através do botão “profile” e dá para ver as diferentes informações pessoais assim como o histórico de alugueres e cupões já utilizados. Nas Promotions/Coupons na homepage vai abrir outro form com as promoções que estão ativas neste momento assim como os cupões. Se clicar nos cupões obtém os cupões e se clicar nas promoções vai para as forms dos alugueres.

Para Testar a aplicação - Como Admin

Fazer login como admin (username:joanadmin@gmail.com, pass: jojona901 de seguida já vai conseguir visualizar o botão para as operações onde pode fazer as operações apresentadas nos botões.

Alteração dos dados

Nota: Para fazer a alteração dos dados em relação ao servidor abra o form authpage e em auth_page_Load() tem la variáveis que pode alterar.