Universidade de Aveiro

Novas Tecnologias da Comunicação

Laboratório Multimédia 4
Mini Projeto Individual
2020

QUINTA DA CARREGOSA
DOURO

email: **admin@admin.pt**
password: **admin**

email: **user@user.pt**
password: **user**

www.quintadacarregosa.pt

Samuel Pinto dos Santos | 89149

O Tema

Sou natural de Tabuaço, um concelho no coração do Douro. Quando li que o tema do trabalho seria livre, automaticamente me lembrei que podia dar ao miniprojecto esta identidade com a qual me relaciono. A empresa de um dos meus tios seria a perfeita oportunidade para pôr em prática esta ideia. Carregosa Vinhos é uma empresa que corre já há algumas gerações na família do meu tio. Este já emprega um dos seus filhos, Fernando Jorge, como enólogo na empresa e têm demonstrado, ao longo destes anos, um enorme sucesso na gestão da empresa.

Contactei então o meu primo para lhe explicar a situação e pedir-lhe autorização para utilizar a empresa e os conteúdos multimédia já existentes no website deles (http://www.carregosavinhos.pt/pt-pt) para a elaboração do miniprojecto no âmbito desta unidade curricular. O mesmo teve a simpatia de me enviar todos os conteúdos novos (ainda não online).

Com estes materiais e toda a informação disponível no seu website, não precisei de me preocupar muito com a pesquisa de informação fidedigna para inserir no projeto. Embora muitos conteúdos, como por exemplo textos descritivos ou especificações técnicas, sejam reais, alguns deles estão repetidos em vários produtos. De notar, também, que os preços não correspondem à realidade.

Dito isto, importa referir que o atual website oficial da empresa não tem um sistema de eCommerce e, por conversas anteriores com o meu primo, sabia que ele estava interessado em proporcionar ao público alvo a possibilidade de compra dos produtos online. Por coincidência, sei que tal projeto já está a ser desenvolvido.

Propus então a mim mesmo fazer todo o sistema de loja online onde fosse possível comprar os produtos da empresa. Para além do que era absolutamente necessário e especificado no enunciado do miniprojecto, os meus objetivos eram:

1. Ter uma lista de todos os produtos disponíveis na loja;
2. Ter uma página individual de produto com os respetivos detalhes;
3. Ter um sistema de carrinho, onde se pudessem adicionar produtos;
4. Ter um sistema de checkout para concluir a compra;
5. Ter uma página de encomendas onde se pudessem ver as compras efetuadas.

Modelo ER e Narrativa de contexto

O contexto do problema e os desafios a solucionar eram os seguintes:

a) A loja online é composta por vinhos com todas as suas especificações;
b) A Carregosa vinhos é detentora de 4 marcas e, por esse motivo, cada produto faz parte de uma das marcas;
c) Cada produto tem associado um estado (Em stock, Limitado, Esgotado);
d) O utilizador pode aceder à página da loja, onde estão todos os produtos e pode adicionar os mesmos ao seu carrinho de compras nas quantidades que entender, consoante o seu estado (se um produto estiver esgotado, não é permitido adicionar o mesmo ao carrinho);
e) Uma vez concluída a compra, registar a lista de produtos, a quantidade dos mesmos, o preço total da encomenda, o estado da mesma (Em processamento, Enviada), e a data em que foi efetuada. Isto para poder criar uma página onde se poderiam ver todas as encomendas feitas pelo utilizador.
f) Já com a compra efetuada, eliminar todos os produtos que antes tínhamos no carrinho, visto que estes já estão registados no histórico de encomendas. É assim possível ao utilizador adicionar outros produtos e, eventualmente fazer outra encomenda.
g) Ser possível ao administrador do website, fazer algum tipo de gestão do mesmo, como por exemplo a promoção de outros membros a administrador ou até a adição de um novo vinho caso se viesse a verificar um acréscimo no número de vinhos oferecidos pela empresa.

Desenvolvi, inicialmente, este modelo lógico (Fig.1). No momento de tentar resolver o acima referido ponto 3, cometi um erro, talvez por falta de atenção. Onde vemos a tabela “status_produto” ligada às tabelas “produtos_encomendas” e “produtos_carrinho” por ligações de N:M, esta apenas deveria estar ligada à tabela “principal_produto_vinho”, onde uma chave estrangeira teria de estar presente. Tal foi corrigido já no momento de desenvolvimento em php em que me deparei com a impossibilidade de restringir a ação de adicionar o produto ao carrinho consoante o seu estado. Por exemplo, se um produto estivesse esgotado, não seria possível adicionar o mesmo ao carrinho.

Embora esta ideia esteja implementada na BD, por falta de tempo e por sentir que tal implementação seria estar a repetir o mesmo tipo de lógica em php, não foi implementada no código. Assim, todos os produtos têm por defeito o valor “Em stock”.

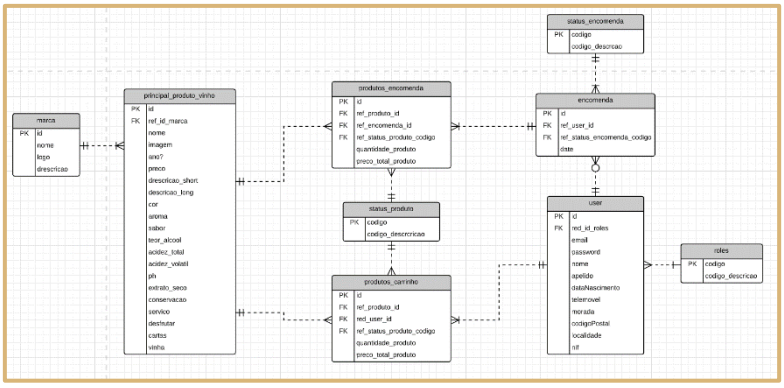

Figura 1 – Base de Dados (Modelo Lógico)

Assim, depois de algumas correções o modelo físico da BD é o seguinte:

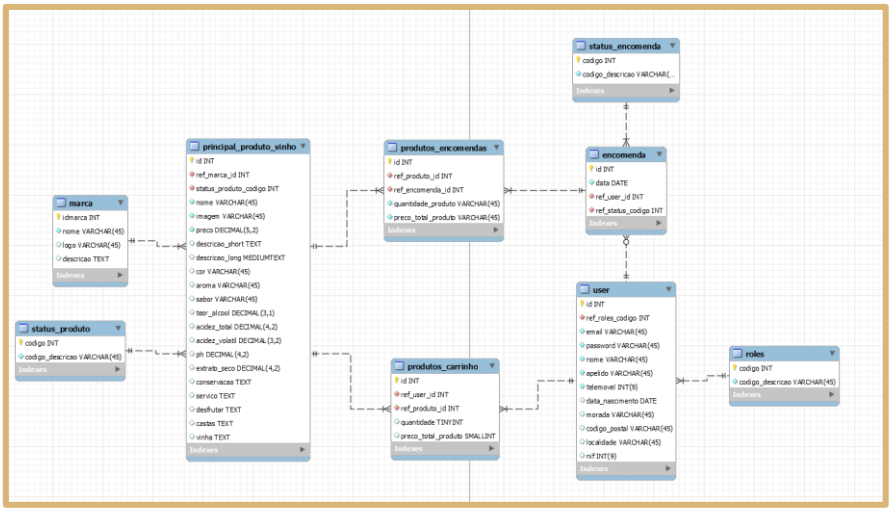

Figura 2 – Base de Dados (Modelo Físico – Segue também em anexo em formato .mwb)

Analisando então o modelo físico (Fig.2) podemos verificar as soluções encontradas para resolver os desafios anteriormente colocados:

a) Visitando o website atual da empresa, recolhi todas as informações necessárias a incluir nos parâmetros da tabela principal, que na minha BD é a tabela dos produtos/vinhos.

b) Entidade "marca", que enumera e contém informação relativa a todas as marcas da empresa.

c) Através de uma ligação de 1:N, a tabela "marca" liga à "principal_xx_xx", permitindo assim que exista uma chave estrangeira nesta última, fazendo com que cada vinho tenha associado uma das marcas.

d) A ação de adicionar um ou mais produtos ao carrinho de compras, traduz-se numa ligação de N:M entre "principal_xx_xx" e "user". Na tabela de relação "produtos_carrinho", irão ficar gravados todos os produtos adicionados a todos os carrinhos dos utilizadores. Obviamente que, um utilizador só consegue ver os produtos que adicionou. Para isso, gravam-se as PK de ambas as tabelas como FK na tabela de relação, assim como a quantidade de um produto que foi adicionado e o preço total desse produto que é calculado através do preço de cada item a multiplicar pela quantidade. O parâmetro do estado do produto provém de uma outra tabela, "status_produto" , que se liga a "principal_xx_xx" através de uma ligação de 1:N, fazendo com que a FK atribua o estado a cada produto.

e) A tabela "produtos_encomendas" tem uma lógica muito semelhante à de carrinho. Esta serve para armazenar os produtos que estão associados à tabela "encomenda". Que, por sua vez, está ligada aos utilizadores. A lógica inerente é que um utilizador possa ter zero ou muitas encomendas, e que cada encomenda necessite de ter um ou mais produtos. Para isso usei relações de 1:N entre "user" e "encomenda"; N:M entre "encomenda" e "principal_xx_xx", dando então origem à tabela de relação "produtos_encomendas". O parâmetro do estado da encomenda provém de uma outra tabela, "status_encomenda" , que se liga a "encomenda" através de uma ligação de 1:N, fazendo com que a FK atribua o estado a cada encomenda.

f) –

g) Para ser possível a distinção entre um utilizador com permissões de administrador e outros com apenas permissões default, criei a tabela "roles", que liga a "user" através de uma ligação 1:N, visto que todos os utilizadores deverão ter associado, através da FK, um role (ou Admin ou User).

Funcionalidades Implementadas

- **Master Detail**

Inicialmente, foquei a minha atenção na página principal, “index.php”. O template já se apresentava com um carrossel que eu achei que se adequaria e iria satisfazer as minhas necessidades: Listar os produtos, com a sua imagem, nome, marca e preço. De notar que todos os parâmetros são da tabela “principal_produto_vinho”, à exceção da marca, que apesar de ter o id como chave primária nessa tabela, o nome correspondente encontra-se na tabela “marca”.

Este carrossel era ótimo pois, da maneira que estava construído, iria permitir-me filtrar os produtos por marca. Para o efeito, usei INNER JOIN. Para fazer o filtro, inicialmente decidi não perder muito tempo a otimizar o código, e por isso deixei o código html original e moldei o php ao mesmo. Isto resultou num código bastante extenso e mais difícil de ler, que pode ser consultado no ficheiro components/cp_all_produts_nao_optimizado.php (este ficheiro não está em uso na versão mais recente do website).

Mais tarde, voltei para otimizar o mesmo que, para ser sincero, foi a parte mais demorada deste projeto, por incrível que pareça. A razão para isto reside no facto de, como precisava de fazer um ciclo “for” de “php” onde necessitava de escrever toda lógica de pedido de informação à base de dados, e durante o ciclo while(fetch), escrever a estrutura correta do carrossel, precisei de algumas horas para perceber o mesmo e de ir resolvendo todos os erros de HTML que iam aparecendo. No entanto, penso que valeu a pena pois, comparando os dois ficheiros, percebe-se que há uma redução notável do código e, consequentemente, é mais simples de ler. Esta versão do código pode ser consultada no ficheiro components/cp_all_produts.php.

Diretamente sobre o master-detail: Com este carrossel a funcionar, a ideia seria clicar num dos produtos e ser redirecionado para uma página de produto individual onde se pudesse consultar mais informação e os detalhes do mesmo. Esta página está desenvolvida no ficheiro single-product.php. Este script é bastante simples recebendo do método GET a variável ‘id_produto’ e, em seguida, efetuando a query de consulta de todos os parâmetros à base de dados. Estes são posteriormente escritos no seu respetivo lugar no código HTML.

Outros exemplos de utilização de INNER JOINS para listar os produtos da tabela principal são os ficheiros : cp_show_cart.php e cp_shop.php.

- **Registo e Login com sessão**

Depois de ter concluído a primeira iteração do carrossel, foquei-me no desenvolvimento do sistema de registo e login. Ambos funcionam, no entanto, no script de registo, achei por bem implementar o código que permitisse ao utilizador, quando se regista com sucesso, ficar automaticamente com a sessão iniciada e não precisar de inserir novamente os seus dados para fazer login. Assim, quando nos registamos no website, e o registo é efetuado com sucesso, somos redirecionados para a página principal, onde podemos ver na navbar, que já está dentro da nossa conta de utilizador. Tive o cuidado, por sugestão do professor, de mudar o nome das variáveis de sessão para prevenir futuros problemas no servidor de Labmm.

- **Gestão de informação**

Agora, já com o sistema de login e registo prontos, pude desenvolver o sistema de gestão de informação dos utilizadores.

Primeiramente, escrevi o código do ficheiro components/cp_alterar_info_user.php que diz respeito a todos os casos, ou seja, independentemente de estarmos com uma conta de administrador ou com uma de utilizador, teremos a possibilidade de alterar os nossos dados. Para isto utilizei um script básico com a utilização de um UPDATE como query de SQL.

Posto isto, e uma vez que na base de dados tenho a diferenciação entre administradores e utilizadores, desenvolvi um sistema que permite aos admins gerirem todas as contas presentes na base de dados. Este sistema permite então, depois de selecionar qual a conta a gerir num <select>, alterar o respetivo email e promover/despromover essa conta a admin/user. Para obrigar o utilizador a inserir a sua palavra pass, como forma de segurança, desenvolvi um modal, em conjunto com uma função no ficheiro scripts/sc_verificar_pass.php. Esta função é chamada no início dos scripts onde os mesmos só correm se esta função retornar o valor true. Para consulta deste código: components/cp_admin_panel.php e scripts/sc_alterar_info_admin.php.

De notar ainda que, depois disto, desenvolvi com a ajuda dos conteúdos disponibilizados pelo professor (nomeadamente https://www.w3schools.com/php/php_file_upload.asp) , o código responsável por, uma vez iniciados com uma conta de admin, ser possível adicionar novos produtos ao website, preenchendo uma série de campos correspondentes aos parâmetros da tabela “principal_produto_vinho”, incluindo o upload de um ficheiro de imagem. Este novo produto será dinâmico, ou seja, uma vez adicionado com sucesso irá aparecer nas outras páginas, juntamente com todos os outros produtos.

- **Carrinho de compras**

Uma vez que tinha a questão do login e registo feita e também a listagem dos produtos pronta, o próximo passo era desenvolver toda a lógica associada ao carrinho de compras. Defini que este só poderia ser acedido após o momento de login. Quando um utilizador clica no botão de adicionar ao carrinho, é lido o ficheiro sripts/sc_add_to_cart.php. Dividi a lógica deste script em três partes:

Primeiro, vamos verificar se o utilizador teve más intenções e tentou aldrabar o website, ou seja, verificar se o item que ele pretende adicionar, realmente existe na nossa base de dados e, no processo, consultar o parâmetro "id" e "preço" do mesmo. Se existir, a variável $produtoExiste toma o valor "true". Caso contrário, "false". Escusado será dizer que o script só continua, caso esta primeira verificação tenha tido sucesso.

Agora, iremos verificar se este produto já existe no nosso carrinho. Porquê? Porque caso já exista, a ideia não é adicionar outra instância na tabela "produtos_carrinho" com o mesmo produto, mas sim mudar o parâmetro de quantidade para serem adicionados quantos produtos o utilizador escolheu. Ou seja, não ter duas instâncias de quantidade 1 do, digamos, produto5 mas sim uma instância com quantidade 2 desse produto. Para isto utilizei um UPDATE como query de SQL. A mesma lógica de variáveis a "true" ou "false" aplica-se nesta fase.

Finalmente, se chegado à fase 3, o script saberá que estamos a adicionar um produto que ainda não existia no carrinho e, por isso, terá de realizar um INSERT novo na tabela "produtos_carrinho".

Agora, faltava apenas fazer um script que permitisse remover um produto do carrinho de compras. Este código está no ficheiro scripts/sc_delete_from_cart.php. Algo bastante simples com recurso a um DELETE como query de SQL.

Desenvolvi, para complementar a usabilidade do website, uma página específica do carrinho, visto que todo este código tinha em vista apenas o carrinho presente na navbar. Esta página foi bastante simples de desenvolver, pois todas as variáveis já estavam à minha disposição, visto que a navbar é um componente que está presente em todas as páginas. Por isso, bastou reutilizar as variáveis e colocá-las numa página HTML diferente. (cart.php)

- **Filtro de marca**

Desenvolvi, posteriormente, um filtro na página shop-grid.php, mais concretamente no componente cp_shop.php. Este tem a seguinte lógica:

Quando se seleciona uma marca no menu <select>, é passado o id dessa marca pelo método GET para a mesma página, ou seja, existe um refresh da página, mas desta vez já com o id da marca presente no URL. Assim, quando a página recarrega, entra numa estrutura de decisão onde verifica se existe ou não o método GET no URL. Caso exista, adapta a query ao id da marca selecionada.

- **Searchbar funcional**

Para além do filtro, escrevi um código bastante simples para o funcionamento da searchbar presente na navbar. Quando escrevermos alguma coisa na mesma, é executada uma query semelhante ao filtro acima explicado. No entanto, neste caso, não se filtram os resultados pela marca, mas sim pelo nome.

Infelizmente, não tive tempo para fazer com que estas duas funcionalidades se interligassem, ou seja, de momento não é possível procurar um produto pelo nome e, por cima, filtrar esses resultados por marca.

Ainda tinha planeado fazer a ordenação dos resultados por preço ou ordem alfabética, no entanto, como já disse, não me foi possível por falta de tempo.

Desafios

Como disse anteriormente, o maior desafio e o que exigiu mais atenção e tempo foi a otimização do script do carrossel na página principal (Fig.3).

Resolvi a questão envolvendo toda a lógica de acesso à base de dados, num ciclo “for” que corre o número de vezes igual ao número de marcas existentes, mais um. Porquê mais um? Porque eu também quero mostrar, na primeira tab do carrossel, todos os produtos, sem filtrar por marca.

Desta forma, quando o ciclo “for” corre pela primeira vez, a variável contador $i é igual a 0 e portanto o script tem um comportamento diferente, executando uma query que seleciona todos os produtos.

Nas próximas iterações do ciclo, o valor de $i muda e a query muda com esse valor. Ou seja, se $i = 1, irá apresentar apenas o resultado dos produtos que têm a marca 1 associada.

Foi esta a maneira que consegui desenvolver. Provavelmente existirá uma maneira mais simples, mas pareceu-me bastante eficaz quando comparada com a solução do script anterior, sem optimização. Este código pode ser consultado no ficheiro components/cp_all_products.php nas linhas 56 a 172.

Figura 3 - Carrossel do index.php

Conclusão

Analisando o resultado, e comparando-o com os objetivos inicialmente definidos, posso verificar que, de facto, não desenvolvi tudo a que me comprometi, nomeadamente a parte de concluir a compra dos produtos e armazenar informação sobre as encomendas. No entanto, esclareço que tal não foi feito, não só por falta de tempo, mas também porque concluí que, o desenvolvimento de tal código seria apenas repetir, de certa forma, a lógica e os mesmos processos até aqui implementados.

Não foi também implementada a restrição de adicionar produtos que tenham o estado de esgotado. Por isso, todos os produtos têm, por default, o estado “Em Stock”.

De notar ainda que, as páginas blog-details são meramente HTML. Resolvi fazer estas páginas para dar a conhecer melhor a empresa em si e também por uma questão de cortesia para com o meu primo e o meu tio que, não só me permitiram fazer este miniprojecto com a empresa deles, como também tiveram a simpatia de me disponibilizar os conteúdos multimédia para o desenvolvimento do mesmo.

Gostaria ainda de dizer que, caso a implementação do website fosse avante, teria de desenvolver estas páginas e rechear o website com toda a informação oficial de todos os produtos.

Por fim, importa concluir que, apesar do tempo limitado, o balanço é muito positivo. Penso que é realmente com estes projetos que desenvolvemos as devidas competências, até mais do que num teste prático. De facto, a realização de um projeto destes obriga o aluno a tentar, a pesquisar soluções e implementá-las corretamente. Na minha modesta opinião, estes trabalhos são de facto uma mais valia para a aprendizagem e preparação dos alunos para o mercado de trabalho onde existem também prazos a cumprir.