

BDAD

Site e-commerce

Bases de Dados

2019/2020

Tiago Lima Rocha – up201406679

António Melo Cabral Lima Braga - up201708995

Carlos Manuel Pires Brás - up201404616

Descrição

Neste trabalho pretende-se ilustrar um modelo de base de dados para um sistema de compra e venda de artigos numa loja online e-commerce, para tal introduzem-se os seguintes termos: Visitante, Utilizador, Administrador, Estado do Utilizador Carrinho de Compras, Item, Produto, Fatura, Encomenda, Estado da Encomenda, Pagamento, PayPal, Cartão de Crédito e Rating, que constituem os elementos (ou entidades) que irão compor este modelo.

Para efeitos de *data-mining*, serão guardadas os endereços de IP de cada visitante (identificador único), bem como a sua localização geográfica, data em que visitou o site e tempo total de visita.

Um visitante do site pode ou não ser um utilizador. Um utilizador pode ser administrador do site.

Cada utilizador tem um id, username (deve ser composto por caracteres alfa-numéricos, com a inclusão de hífenes e underscores; não poderá conter caracteres especiais ou espaços), password (deverá ter um tamanho maior ou igual a 16 caracteres, ter pelo menos uma letra maiúscula, uma minúscula, um algarismo e um carácter especial), e-mail (deverá seguir o seguinte formato: exemplo@foo.bar), nome, NIF, morada, código postal, data de nascimento e género.

Administrador é uma especialização de utilizador com privilégios especiais de edição de produtos, utilizadores ou até mesmo de alguns elementos do *frontend* do próprio site.

O Utilizador tem um estado que dita se este está atualmente ativo, ou seja “*logged in*”, permanentemente banido ou temporariamente suspenso do site. Só será possível efetuar encomendas se o seu estado for “ativo”.

Cada conta terá um carrinho de compras que possuirá um identificador único, uma data de criação e uma data de eliminação. Este servirá como uma “*wish-list*” cujo verdadeiro propósito será guardar informação para data-mining, como hábitos de procura, tendências, etc. Cada carrinho possuirá itens que serão referências para os produtos que este deseja.

Produtos do site terão um id único, devem pertencer a uma categoria de produtos (p. ex. Alimentares, higiene, escritório, etc), terão também um nome e um preço por unidade,

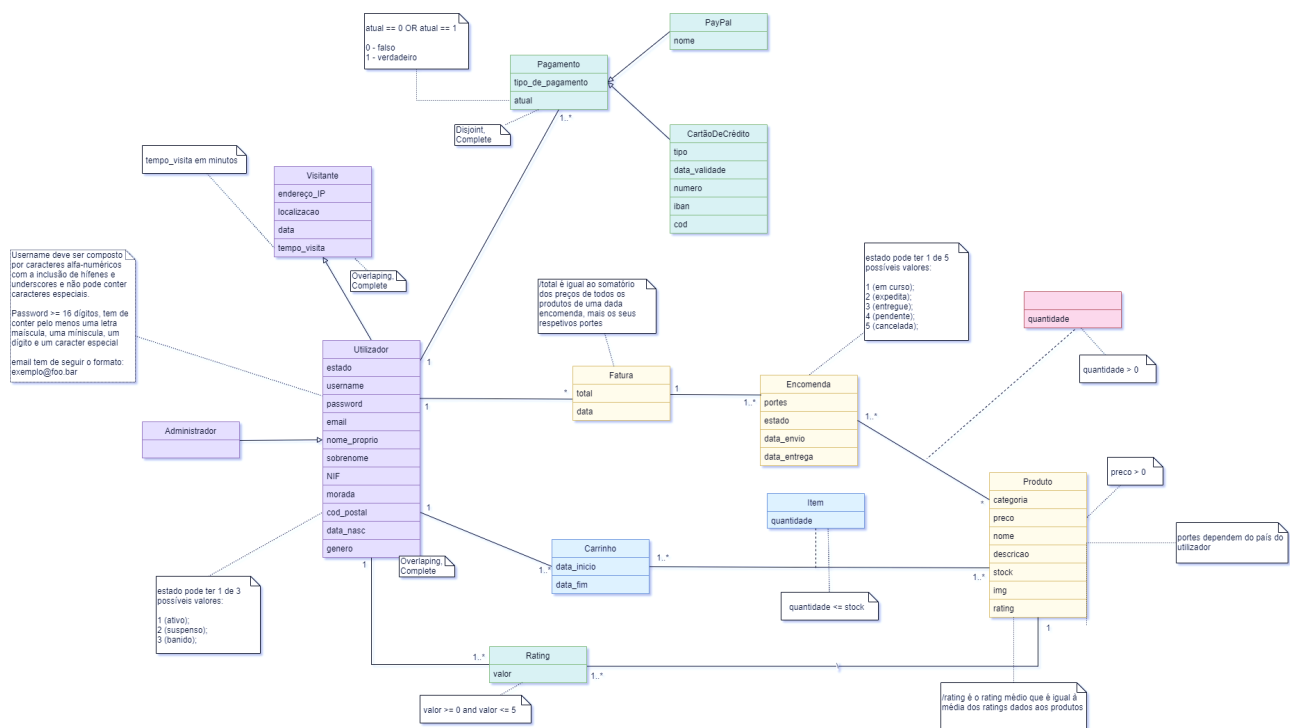
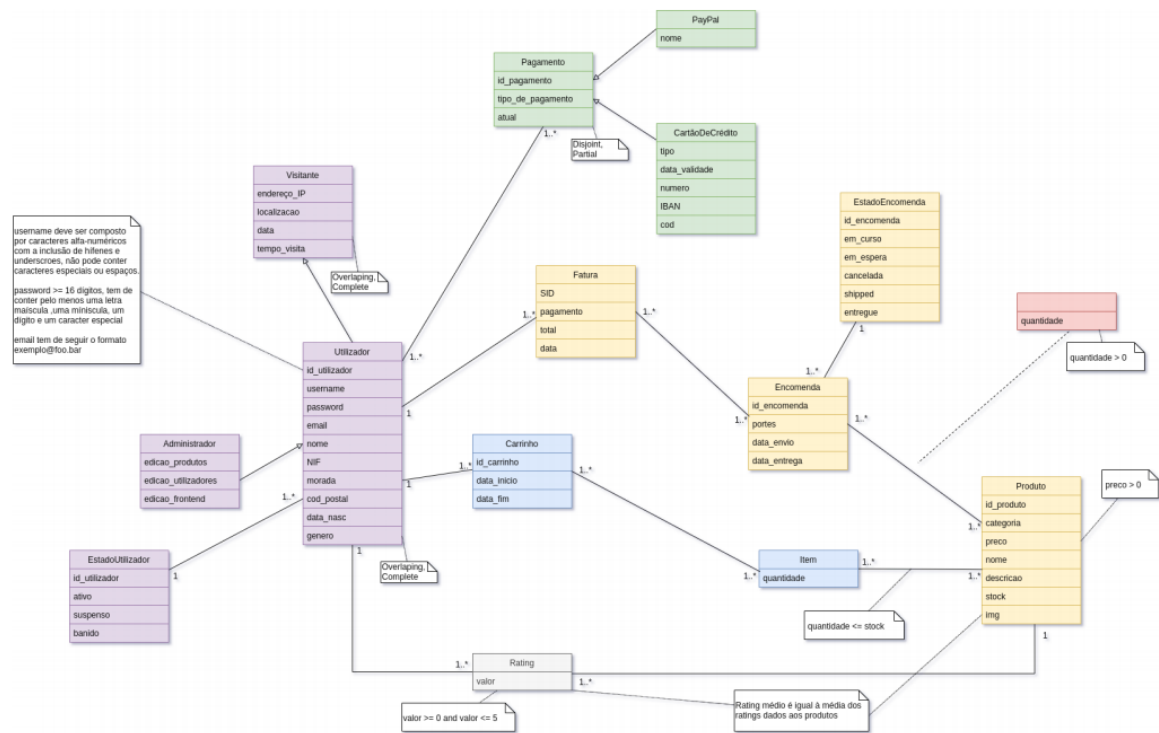
uma descrição, uma imagem e deverá ser possível a qualquer altura saber a quantidade de um dado produto disponível em stock. A cada produto estará associado um *rating* médio dado pelos utilizadores do site.

As encomendas estão sempre associadas a faturas e em cada encomenda estarão associados um ou mais produtos. Estas terão um identificador único e o custo dos portes associados, bem como uma data de envio e uma data de entrega. Para seguir o progresso de cada encomenda, esta terá um estado que irá atualizando ao longo das várias etapas de entrega. Assim será sempre possível para cada utilizador seguir a sua encomenda.

A fatura que estará também associada a um utilizador (embora um utilizador possa ter mais do que uma fatura registada) terá também um identificador único, deverá nela ser declarado o método de pagamento utilizado, bem como o montante final em euros mais uma data de emissão.

Quanto aos tipos de pagamento, cada utilizador poderá ter mais do que um tipo de pagamento registado na sua conta, porém apenas um será marcado como o atual método usado numa dada compra (podendo o utilizador escolher a qualquer altura qual o método desejado para efetuar compras). O utilizador poderá então pagar com PayPal se desejar ou com cartões de crédito (Visa, MasterCard, etc).

Modelo Conceptual



Modelo Relacional

Visitante (endereco_IP, localizacao, data, tempo_visita)

- Chave Primária: endereco_IP
- Chave Estrangeira: N/A

Utilizador (id_utilizador, estado, username, password, email, nome proprio, sobrenome, nif, morada, cod_postal, data_nasc, genero, endereco_IP → visitante)

- Chave Primária: id_utilizador
- Chave Estrangeira: endereco_IP

Administrador (id_utilizador → utilizador)

- Chave Primária: id_utilizador
- Chave Estrangeira: N/A

Carrinho (id_carrinho, data_inicio, data_fim, id_utilizador → Utilizador)

- Chave Primária: id_carrinho
- Chave Estrangeira: id_utilizador

Produto (id_produto, categoria, nome, descricao, stock, img, preco, portes, rating)

- Chave Primária: id_produto
- Chave Estrangeira: N/A

Rating (id_rating, valor, id_utilizador → Utilizador, id_produto → Produto)

- Chave Primária: id_rating
- Chave Estrangeira: id_utilizador, id_produto

Encomenda (id_encomenda, portes, estado, data_envio, data_entrega, id_fatura → Fatura)

- Chave Primária: id_encomenda
- Chave Estrangeira: id_fatura

Fatura (id_fatura, pagamento, total, data, id_utilizador → Utilizador)

- Chave Primária: id_fatura
- Chave Estrangeira: id_utilizador

Pagamento (id_pagamento, tipo_de_pagamento, atual, id_utilizador → Utilizador)

- Chave Primária: id_pagamento
- Chave Estrangeira: id_utilizador

PayPal (id_pagamento → Pagamento, nome)

- Chave Primária: id_pagamento
- Chave Estrangeira: N/A

CartaoDeCredito (id_pagamento → Pagamento, tipo, data_validade, numero, iban, cod)

- Chave Primária: id_pagamento
- Chave Estrangeira: N/A

ProdutoEncomenda (id_encomenda → Encomenda, id_produto → Produto, quantidade)

- Chave Primária: id_encomenda, id_produto
- Chave Estrangeira: id_encomenda, id_produto

ProdutoCarrinho (id_carrinho → Carrinho, id_produto → Produto, quantidade)

- Chave Primária: id_carrinho, id_produto
- Chave Estrangeira: id_carrinho, id_produto

Dependências Funcionais e Análise da Forma Normal

Dependências Funcionais

Visitante:

$\{\text{endereco_IP}\}^+ = \{\text{endereco_IP}, \text{localizacao}, \text{data}, \text{tempo_visita}\}$

Utilizador:

$\{\text{id_utilizador}\}^+ = \{\text{id_utilizador}, \text{estado}, \text{username}, \text{password}, \text{email}, \text{nome_proprio}, \text{sobrenome}, \text{nif}, \text{morada}, \text{cod_postal}, \text{data_nasc}, \text{género}, \text{endereço_IP}\}$

Administrador:

$\{\text{id_utilizador}\}^+ = \{\text{id_utilizador}\}$

Carrinho:

$\{\text{id_carrinho}\}^+ = \{\text{id_carrinho}, \text{data_inicio}, \text{data_fim}, \text{id_utilizador}\}$

Produto:

$\{\text{id_produto}\}^+ = \{\text{id_produto}, \text{categoria}, \text{nome}, \text{descricao}, \text{stock}, \text{img}, \text{preco}, \text{rating}\}$

Rating:

$\{\text{id_rating}\}^+ = \{\text{id_rating}, \text{valor}, \text{id_utilizador}, \text{id_produto}\}$

Encomenda:

$\{\text{id_encomenda}\}^+ = \{\text{id_encomenda}, \text{portes}, \text{estado}, \text{data_envio}, \text{data_entrega}, \text{id_fatura}\}$

Fatura:

$\{id_fatura\}^+ = \{id_fatura, pagamento, total, data, id_utilizador\}$

Pagamento:

$\{id_pagamento\}^+ = \{id_pagamento, tipo_de_pagamento, atual, id_utilizador\}$

Paypal:

$\{id_pagamento\}^+ = \{id_pagamento, nome\}$

CartãoDeCredito:

$\{id_pagamento\}^+ = \{id_pagamento, tipo, data_validade, numero, iban, cod\}$

ProdutoEncomenda:

$\{id_encomenda, id_produto\}^+ = \{id_encomenda, id_produto, quantidade\}$

ProdutoCarrinho:

$\{id_carrinho, id_produto\}^+ = \{id_carrinho, id_produto, quantidade\}$

Análise da forma normal:

Para uma dependência funcional estar na Boyce Codd Normal Form (BCNF), o lado esquerdo da dependência tem de corresponder a uma key para a relação. Para fazer a análise da 3ª Forma Normal (3NF), podemos assumir que se a dependência se encontra na BCNF então também tem de estar na 3NF, pois a 3NF é um superset da BCNF. Como podemos ver, o lado esquerdo é uma key em todas as dependências funcionais acima, logo podemos concluir que as dependências funcionais estão na BCNF, e consequentemente na 3NF.

Restrições à Base de Dados

Visitante

- endereco_IP deve ser único;
- tempo_visita deve ser positivo;

Utilizador

- estado tem de ser 1, 2 ou 3;
- cod_postal deve ter um tamanho igual a 8;
- genero deve ser ou 0 ou 1;
- username deve ser composto por caracteres alfa-numéricos com a inclusão de hífen e underscores e não pode conter caracteres especiais
- password tem de ter 16 ou mais dígitos, tem de conter pelo menos uma letra maiúscula, uma minúscula, um dígito e um carácter especial;
- email tem de seguir o formato: exemplo@foo.bar;

Produto

- preco deve ser maior que zero;
- rating tem de ser maior ou igual a zero e menor ou igual a 5;

Rating

- valor tem de ser maior ou igual a zero e menor ou igual a 5;

Encomenda

- estado tem de ser maior ou igual a zero e menor ou igual a 5;
- data_entrega deve ser maior que data_envio;

Encomenda

- numero tem de ter um tamanho inferior a 20 digitos;
- cod tem de ser menor que 999;

Carrinho

- Sempre que o estado de um utilizador muda para 1 (ativo) é criado um novo carrinho;

ProdutoCarrinho

- quantidade tem de ser maior que o stock do produto

Interrogação da Base de Dados

Interrogação 1:

Selecionar todos os produtos (e toda a informação a eles referente) de uma dada categoria (i.e. "Fotografia, Video, Lab Foto")

Interrogação 2:

Selecionar o valor de todas as compras efetuadas no dia atual por um determinado utilizador e o seu respetivo nome, morada e email

Interrogação 3:

Selecionar todas as encomendas entregues do último mês de uma dada categoria (i.e.), cuja média dos produtos é superior a X (i.e. 3), interrogando quais os nomes dos produtos, respetivos IDs e rating, e o total pago em cada encomenda.

Interrogação 4:

Selecionar os 10 produtos mais vendidos no último mês, respetivos nomes, IDs e categorias, bem como os seus preços e rating.

Interrogação 5:

Selecionar número de produtos vendidos na última semana e o respetivo valor total de receita gerada.

Interrogação 6:

Selecionar o número total de vendas efetuadas no último mês e o total de receitas geradas.

Interrogação 7:

Selecionar todos os produtos que foram comprados quando um determinado produto foi vendido (i.e. Produto com ID 1234).

Interrogação 8:

Selecionar o número de produtos de cada categoria com um rating superior a X (i.e. 3,5).

Interrogação 9:

Selecionar o número total de encomendas efectuadas no dia atual.

Interrogação 10:

Selecionar o tempo médio de visita no site.

Adição de gatilhos à base de dados

Gatilho 1:

Sempre que um utilizador fica online cria um novo carrinho para esse utilizador se um já não existia previamente.

Gatilho 2:

Sempre que se regista um novo utilizador cria um carrinho para esse utilizador.

Gatilho 3:

Como a tabela `Utilizador` contém informação sensível que não pode ser perdida, é importante armazenar esta informação numa tabela separada `UtilizadorLogs` que contém os 'logs' após cada alteração, para isso usamos este trigger.

Gatilho 4:

Atualiza automaticamente o valor total a pagar na fatura sempre que é dada uma nova entrada na tabela `ProdutoEncomenda`.

Gatilho 5:

Verifica se a quantidade selecionada para um dado produto colocado no carrinho é inferior à quantidade desse produto existente em stock. Se não for, atira um alerta.

Gatilho 6:

Verifica se o email inserido aquando do registo de um novo usuário no site contém um formato de email válido. Se não, atira um alerta.

Gatilho 7:

Possível implementação para uma verificação da validade da palavra-passe e do nome de utilizador aquando de um novo registo de usuário no site, usando regex. A implementação da função REGEXP é feita neste caso em Java, mas poderia ter sido usada outra linguagem, por exemplo, Python ou PHP. Verifica se quer a palavra-passe, quer o nome de utilizador seguem as normas estabelecidas pelo padrão regex.

Gatilho 8:

No caso da tabela Produto-Carrinho é interessante saber quais os produtos que os utilizadores decidiram remover do carrinho, sempre que isso acontece guarda essa informação numa tabela `ProdutoCarrinhoLogs`.