

## Projeto de BD – Parte 3

Grupo 12

Turno BD2L06

Professores **Pedro Sousa** e **João Caldeira**

Aluno	Esforço (horas)	Esforço Relativo
João Amadeu (98943)	14 horas	33%
Mateus Correia (103557)	14 horas	33%
João Iria (98954)	14 horas	33%

### 1) Desenvolvimento da Aplicação

A aplicação foi concebida com o auxílio das linguagens HTML, CSS, Python e JavaScript. O site consiste em uma plataforma para gerenciamento de produtos, fornecedores, clientes e pedidos. O objetivo principal é facilitar o processo de registo, remoção e alteração de informações relacionadas a essas entidades.

O desenvolvimento da aplicação envolveu a criação de diferentes páginas HTML para cada funcionalidade. O HTML foi utilizado para estruturar o conteúdo das páginas e definir os elementos interativos, como formulários e botões. O CSS foi utilizado para estilizar as páginas, garantindo uma experiência visual agradável e consistente.

A interatividade e dinamismo da aplicação foram implementados por meio do JavaScript. Através dele, foi possível criar funções para manipular eventos, como cliques e submissões de formulários, e atualizar o conteúdo das páginas de forma assíncrona.

No lado do servidor, a linguagem Python foi utilizada para processar as requisições feitas pelos usuários e realizar as operações de registo, remoção e alteração de dados no banco de dados. O Python também foi responsável por estabelecer a conexão com o banco de dados e executar consultas SQL para obter e modificar as informações necessárias.

Além disso, foi realizado o uso de índices no banco de dados para otimizar o desempenho das consultas mais frequentes. Os índices foram criados em atributos relevantes para as consultas, permitindo uma busca mais eficiente e reduzindo o tempo de resposta.

No geral, o desenvolvimento da aplicação envolveu a integração harmoniosa de HTML, CSS, Python e JavaScript para criar uma plataforma intuitiva e eficiente para o gerenciamento de produtos, fornecedores, clientes e pedidos. A combinação dessas tecnologias proporcionou uma experiência interativa ao usuário e melhorou a performance das consultas e operações realizadas no sistema.

## **2) Funcionalidades da Aplicação:**

A aplicação oferece várias funcionalidades para facilitar o gerenciamento de dados relacionados a produtos, fornecedores, clientes e pedidos. O site tem botões em todas as páginas para *submit*, ("submit" no caso de implementação, e "remove" no caso de remoção, "pay" no caso de pagamento, e "change" no caso de alteração). Abaixo estão descritas as principais funcionalidades disponíveis:

- **Registo de Clientes (Register Client):**

A funcionalidade "Register Client" permite o cadastro de novos clientes na aplicação. Os usuários podem preencher informações como número do cliente (ID), nome, e-mail(único), telefone e endereço (formato "rua XXXX-XXX cidade). Esses detalhes são essenciais para manter um registo completo dos clientes e facilitar a comunicação e o envio de informações relevantes.

- **Remoção de Clientes (Remove Client):**

Com a funcionalidade "Remove Client", é possível excluir clientes do sistema. Os usuários podem inserir o número do cliente (ID) a ser removido e, em seguida, realizar a ação. Essa funcionalidade é útil quando um cliente não é mais relevante ou não deseja mais utilizar os serviços da empresa.

- **Registo de Produtos (Register Product):**

A funcionalidade "Register Product" permite o cadastro de novos produtos na aplicação. Os usuários podem fornecer informações como SKU (código do produto), nome, descrição, preço (até dois caracteres decimais) e EAN (número de artigo europeu). Esses detalhes são importantes para manter um inventário atualizado e fornecer informações detalhadas sobre os produtos disponíveis.

- **Remoção de Produtos (Remove Product):**

A funcionalidade "Remove Product" permite a exclusão de produtos do sistema. Os usuários precisam fornecer o SKU do produto a ser removido para realizar a ação. Isso é útil quando um produto não está mais disponível ou não é mais comercializado.

- **Registo de Fornecedores (Register Supplier):**

Com a funcionalidade "Register Supplier", é possível cadastrar novos fornecedores na aplicação. Os usuários podem inserir detalhes como SKU do fornecedor, nome, endereço (formato "rua XXXX-XXX cidade) e SKU do produto fornecido. Isso ajuda a manter um registo dos fornecedores e a estabelecer uma relação clara entre eles e os produtos fornecidos.

- **Remoção de Fornecedores (Remove Supplier):**

A funcionalidade "Remove Supplier" permite a exclusão de fornecedores do sistema. Os usuários devem fornecer o SKU do fornecedor a ser removido para realizar a ação. Isso é útil quando um fornecedor não está mais ativo ou não fornece mais produtos à empresa.

- **Realização de Pedidos (Make Order):**

A funcionalidade "Make Order" permite a criação de novos pedidos. Os usuários podem inserir informações como ID do pedido, ID do cliente, SKU do produto, data do pedido e quantidade desejada. Isso permite registrar as transações de compra e manter um histórico de pedidos para referência futura.

- **Modificação de Produto (Change Product)**

Permite a modificação da descrição e preço de um produto existente com base no SKU do produto.

- **Pagamento de Pedidos (Pay Order):**

Com a funcionalidade "Pay Order", os usuários podem efetuar o pagamento de pedidos pendentes. Basta fornecer o ID do pedido e o ID do cliente para realizar o pagamento. Isso ajuda a manter um registro dos pagamentos recebidos e a acompanhar o status dos pedidos.

### 3) LINK:

<https://web2.tecnico.ulisboa.pt/ist1103557/website.html>

O Site encontra-se na WEB do sigma do aluno Mateus Correia.

### 4) Índices:

#### 7.1)

```
CREATE INDEX idx_price_date ON orders (price, date);
```

Vantagens:

**Índice idx\_order\_date:**

O índice idx\_order\_date, criado na tabela "orders" com base no campo "date", traz diversas vantagens para a consulta. Ao criar esse índice, obtemos melhorias significativas no desempenho e na eficiência das consultas que envolvem a data dos pedidos. As principais vantagens são:

1. Melhoria na velocidade de busca: O índice permite que o banco de dados encontre rapidamente os registos com base na data, reduzindo o tempo necessário para realizar a consulta. Isso é especialmente útil quando a tabela possui um grande volume de dados.

2. Otimização de consultas com filtros de data: Com o índice, as consultas que envolvem a extração do ano da data, como no exemplo em que se busca pelo ano de 2023, podem ser executadas de forma mais eficiente. O índice ajuda o banco de dados a localizar os registos relevantes sem percorrer todos os registos da tabela.
3. Aumento na eficiência de consultas conjuntas: Quando a tabela "orders" é usada em junções com outras tabelas, o índice idx\_order\_date pode ser aproveitado para melhorar o desempenho geral da consulta. O banco de dados pode usar o índice para identificar rapidamente os registos relevantes e realizar as operações de junção de maneira mais eficiente.

### **Índice idx\_product\_price:**

O índice idx\_product\_price, criado na tabela "product" com base no campo "price", também oferece vantagens significativas para a consulta. As vantagens desse índice são:

1. Agilidade na busca por preço: Com o índice criado no campo "price", as consultas que envolvem filtros de preço, como "price > 50", podem ser executadas de forma mais rápida. O índice ajuda o banco de dados a localizar os registos com o preço desejado de maneira eficiente, reduzindo o tempo de resposta da consulta.
2. Otimização de consultas de agrupamento: O índice idx\_product\_price pode ser utilizado em consultas que envolvem agrupamentos, como no exemplo em que se calcula a soma do preço multiplicado pela quantidade agrupado por número do pedido. O índice permite que o banco de dados acesse os registos de forma eficiente para realizar os cálculos necessários.

Melhoria no desempenho geral: Ao criar o índice no campo "price", a performance geral das consultas que envolvem a tabela "product" é aprimorada. O banco de dados pode utilizar o índice para otimizar as operações de busca e junção, resultando em tempos de resposta mais curtos e um processamento mais eficiente.

## **7.2 )**

-- Index for product table

```
CREATE INDEX idx_product_name ON product (name);
```

-- Index for contains table

```
CREATE INDEX idx_contains_sku ON contains (SKU);
```

Vantagens:

### **Índice idx\_product\_name:**

O índice idx\_product\_name, criado na tabela "product" com base no campo "name", traz diversas vantagens para a consulta. As principais vantagens são:

1. Melhoria no desempenho das consultas com filtro de nome: Com o índice, as consultas que envolvem filtros de nome, como no exemplo em que se busca por nomes que começam com "A%", podem ser executadas de forma mais rápida. O índice ajuda o banco de dados a localizar os registos com base no nome de maneira eficiente, reduzindo o tempo de resposta da consulta.

2. Otimização de consultas de junção: Quando a tabela "product" é usada em junções com outras tabelas, o índice idx\_product\_name pode ser aproveitado para melhorar o desempenho geral da consulta. O banco de dados pode usar o índice para identificar rapidamente os registos relevantes com base no nome e realizar as operações de junção de maneira mais eficiente.
3. Aumento na eficiência de consultas de agrupamento: O índice pode ser utilizado em consultas que envolvem agrupamentos, como no exemplo em que se calcula a soma do preço multiplicado pela quantidade agrupado por número do pedido. O índice permite que o banco de dados ascenda os registos de forma eficiente para realizar os cálculos necessários, melhorando o desempenho dessas consultas.

### **Índice idx\_contains\_sku:**

O índice idx\_contains\_sku, criado na tabela "contains" com base no campo "SKU", também oferece vantagens significativas para a consulta. As vantagens desse índice são:

1. Agilidade na busca por SKU: Com o índice criado no campo "SKU", as consultas que envolvem a busca por SKU podem ser executadas de forma mais rápida. O índice ajuda o banco de dados a localizar os registos com o SKU desejado de maneira eficiente, reduzindo o tempo de resposta da consulta.
2. Otimização de consultas de junção: Quando a tabela "contains" é usada em junções com outras tabelas, o índice idx\_contains\_sku pode ser utilizado para melhorar o desempenho geral da consulta. O banco de dados pode usar o índice para identificar rapidamente os registos relevantes com base no SKU e realizar as operações de junção de maneira mais eficiente.

Em resumo, os índices criados trazem melhorias no desempenho das consultas, permitindo que elas sejam executadas de forma mais rápida e eficiente. Isso resulta em uma aplicação mais ágil e responsiva, proporcionando uma melhor experiência para os usuários.