

## CRIAÇÃO DE UM INVENTÁRIO DIGITAL PARA ESCOLA



**Aluno:** André Filipe Lopes Damasceno

**Número:** a6419



Agrupamento de Escolas de Pinhel

Curso Profissional de Técnico de Multimédia

2018/2021



## CRIAÇÃO DE UM INVENTÁRIO DIGITAL PARA A ESCOLA

### PROVA DE APTIDÃO PROFISSIONAL

**Aluno:** André Filipe Lopes Damasceno

**Número:** a6419

**Curso:** Técnico de Multimédia

**Diretor de Turma:** António Marques

**Coordenadora do Curso:** Ana Lourenço

## Agradecimentos

Antes de mais e em primeiro lugar, quero agradecer aos meus pais, por estarem sempre a motivar-me e a darem-me o apoio que eu necessito.

Agradeço também a todos os meus colegas de turma, por todos os momentos que vivemos juntos durante estes três anos.

A todos os professores por quem passei desde o décimo até ao décimo segundo ano.

Ao professor e Diretor de Turma, Paulo Monteiro, pelo apoio e a motivação extra que ele nos conseguia dar, tentando sempre manter um relacionamento amigável entre alunos e professores, Muito obrigado professor Paulo!

Agradeço também ao professor António Marques, por me mostrar como pode ser difícil e percebendo como vai ser um mercado de trabalho embora não muito bom às vezes podendo dar a volta por cima.

Agradeço também à coordenadora dos cursos e também professora de Português, Ana Lourenço, que fez com que nada nos faltasse durante estes três anos, um muito obrigado!

Por fim, mas não menos importante, agradeço ao senhor Diretor do Agrupamento, Dr. José Monteiro Vaz, por todo o apoio que me deu, ao longo do curso e pelo voto de confiança pensando sempre nos alunos e no seu bem-estar a nível de educação escolar.

## Resumo

De forma geral, a programação é um processo de escrita, testes e manutenção de programas de computadores. Esses programas, por sua vez, são compostos por conjuntos de instruções determinados pelo programador que descrevem tarefas a serem realizadas pela máquina e atendem diversas finalidades.

O meu gosto pela programação surgiu nos primeiros dias de aulas de Sistemas de Informação no 10º ano. Até ao dia a palavra programação, para mim, era praticamente desconhecida. No início até achei um pouco complicado, mas eu sentia um gosto por aquela dificuldade, e com a ávida do professor que naquele tempo era o professor Paulo. Inicialmente achei difícil, porém com o passar do tempo comecei a ganhar gosto pela programação.

Surgiu a Prova de Aptidão Profissional, e como projeto decidi fazer algo que envolvesse programação como também queria ajudar a escola ao máximo porque apesar de tudo nunca fui um aluno fácil de lidar, então decidi criar uma base de dados digital para facilitar algum trabalho à escola como também aprimorar o desenvolvimento da escola.

## Palavras-Chave

Programação; IntelliJ; Dreamweaver; Inventário

## Índice

Capítulo I – Introdução.....	6
1.1. Introdução .....	7
1.2. Escolha do Projeto .....	8
Capítulo II - Enquadramento Teórico .....	9
2.1. O Conceito de Programação .....	10
2.2. A História da Programação .....	11
2.3. Hardwares, softwares e programação .....	12
2.4. Linguagem de Programação .....	13
2.5. Código Fonte.....	13
2.6. Habilidades de um programador .....	14
2.7. A programação e o avanço das Tecnologias Digitais .....	15
2.8. O futuro da programação.....	16
2.9. Cuidados ao criar uma base de dados.....	22
2.9.1. Segmentar as informações.....	22
2.9.2. Definir os índices e relações.....	23
2.9.3. Preservar a cronologia das informações.....	23
2.9.4. Priorizar a integridade dos dados .....	23
2.9.5. Definir os tipos de dados.....	24
2.10. História das Bases de Dados.....	25
Capítulo III – Programas para criar sites .....	31
3.1. Dreamweaver .....	32
3.2. Notepad++.....	34
3.3. Python .....	35

Capítulo IV – Desenvolvimento do Projeto .....	36
4.1. Aplicação utilizada.....	37
4.2. Barras de Ferramentas e Algumas Funcionalidades .....	37
Capítulo V – Conclusões .....	41
5.1. Análise Crítica.....	42
5.2. Conclusão .....	43
Webgrafia .....	44

## Índice de Figuras

Figura 1 - Máquina Diferencial, ou Máquina Analítica – criada por Charles Babbage (1791 – 1871) .....	10
Figura 2 – Processo de guardar e obter dados.....	25
Figura 3 - SGBD.....	26
Figura 4 - Charles Bachman.....	27
Figura 5 – Modelo Relacional.....	28
Figura 6 -SQL .....	29
Figura 7 – Logotipo do Notepad++.....	34
Figura 8 – Logotipo do Python .....	35
Figura 9 - Ambiente de trabalho do IntelliJ.....	37

## Capítulo I – Introdução

## 1.1. Introdução

A Prova de Aptidão Profissional faz parte integrante do curso e deve ser um projeto transdisciplinar integrador de todos os saberes e capacidades que foram adquiridas ao longo dos 3 anos.

Atualmente, a tecnologia desenvolvida nesta área, permite criar aplicações muito facilmente.

O meu projeto tem como objectivo criar um inventário digital para facilitar a inserção de material e a revisão de material em tempo real mantendo informado o grupo de informática sobre a quantidade, características e estado de todo o material informático.

O inventário digital veio ajudar e substituir o inventário a papel facilitando o controlo do material “emprestado, danificado ou em reparação,” tendo assim um melhor controlo do material existente.

A elaboração deste projeto requereu um trabalho constante ao longo de todo o ano, que se iniciou com uma pesquisa aprofundada sobre o tema, a qual faz parte deste documento.

## 1.2. Escolha do Projeto

Ao longo do curso foram exploradas várias ferramentas nas diferentes áreas da multimédia, no entanto a que me despertou maior interesse foi sem dúvida a programação.

Neste sentido, a minha ideia de PAP surgiu quando, na disciplina de SI, estávamos a iniciar a criação de sites e de base de dados, pensei então em criar uma base de dados do material da escola para facilitar a vida à escola que tanto me ajudou.

Assim, a vontade de explorar este tipo de ferramentas, aprofundando os meus conhecimentos numa área que está cada vez mais presente na sociedade, juntamente com o apoio dos meus professores e colegas levaram à concretização deste projeto.

## Capítulo II - Enquadramento Teórico

## 2.1. O Conceito de Programação

De forma geral, a programação é um processo de escrita, testes e manutenção de programas de computadores. Esses programas, por sua vez, são compostos por conjuntos de instruções determinados pelo programador que descrevem tarefas a serem realizadas pela máquina e atendem diversas finalidades.

O advento da programação está diretamente ligado à história dos computadores. A primeira programadora foi uma mulher chamada Ada Lovelace, que escreveu um código para possibilitar a utilização da máquina analítica de Charles Babbage, uma máquina robusta, de difícil comunicação, considerada a precursora dos computadores eletrónicos atuais.

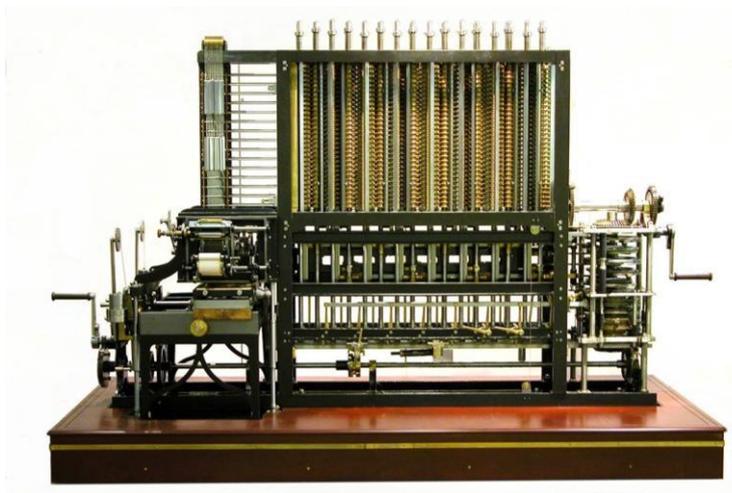


Figura 1 - Máquina Diferencial, ou Máquina Analítica – criada por Charles Babbage (1791 – 1871)

Um século depois, os computadores ficaram cada vez mais fáceis de serem utilizados e a programação ganha cada vez mais espaço na criação dessas tecnologias. Mas, para compreender de verdade do que se trata, é importante conhecer dois elementos fundamentais que compõem todos os computadores modernos, que são o *hardware* e o *software*.

## 2.2. A História da Programação

O primeiro trabalho de linguagem de programação foi criado por Ada Lovelace, grande amiga de Charles Babbage. O projeto da primeira calculadora mecânica programável foi idealizado por Charles Babbage que, após gastar fortunas e um longo tempo, não conseguiu concretizar o projeto. A linguagem de programação ADA foi batizada em homenagem a esta primeira programadora.

Uma das primeiras linguagens de programação para computadores foi provavelmente Plankalkül, criada por Konrad Zuse na Alemanha Nazista, mas que teve pouco ou nenhum impacto no futuro das linguagens de programação.

O primeiro compilador foi escrito por Grace Hopper, em 1952, para a linguagem de programação A-0. A primeira linguagem de programação de alto nível amplamente usada foi Fortran, criada em 1954. Em 1957 foi criada B-0, sucessora da A-0, que daria origem a Flow-Matic (1958), antecessor imediato de COBOL, de 1959. O COBOL foi uma linguagem de ampla aceitação para uso comercial. A linguagem ALGOL foi criada em 1958-1960. O ALGOL-60 teve grande influência no projeto de muitas linguagens posteriores.

A linguagem Lisp foi criada em 1958 e tornou-se amplamente utilizada na pesquisa na área de ciência da computação mais proeminentemente na área de Inteligência Artificial. Outra linguagem relacionada ao campo da IA que surge em 1972 é a linguagem Prolog, uma linguagem do paradigma lógico.

A orientação a objetos é outro marco importante na história das linguagens de programação. A linguagem Simula 67 introduz o conceito de classes. A linguagem Smalltalk expande o conceito de classes e se torna a primeira linguagem de programação que oferecia suporte completo à programação orientada a objetos. A linguagem C++ (originalmente conhecida como C com classes) populariza a orientação a objetos.

## 2.3. Hardwares, softwares e programação

Um computador é formado por componentes físicos, como a parte externa do seu celular, a CPU de um PC, a memória, o rato, enfim... os meios pelos quais os sinais elétricos podem ser convertidos em dados, informações.

Esses componentes são chamados de *hardware* e a sua linguagem binária, que é a linguagem de máquina, é composta apenas por bits, que são zeros e uns. Os bits representam a ausência ou presença de sinais elétricos. Agora, imagine utilizar o seu telemóvel tendo que interpretar informações binárias: se tudo o que se vê na tela fosse uma sequência infinita de zeros e uns, os telemóveis não seriam nada práticos, não é? É por causa disso que existem os softwares.

O software é o meio pelo qual a linguagem de máquina pode ser compilada ou interpretada, através de códigos criados em uma linguagem intermediária, para idiomas que conhecemos, como o português, o inglês e também para imagens, cores, números... enfim, uma série de dados que podem ser assimilados mais facilmente pelos seres humanos.

Alguns exemplos de softwares são os programas de computadores, como o Word, Excel e Paint, e as aplicações para telemóveis, como a Playstore, App Store ou o Uber. Os sistemas operativos, como o Android, Windows, Linux ou o Mac, também podem ser vistos, de forma simplificada, como softwares gigantes que comportam outros softwares.

Mas, afinal, qual é a relação entre hardware, software e programação? A programação é exatamente quem possibilita a existência dos softwares e, por consequência, a utilização mais prática dos hardwares. Para poder dar origem aos softwares, a programação ganha uma linguagem própria que compõe códigos escritos por programadores.

## 2.4. Linguagem de Programação

A linguagem de programação nada mais é do que um método padronizado que permite comunicar instruções específicas para um computador/telemóvel. Há centenas de variações e todas elas permitem que o programador defina sobre quais os dados que o computador deverá atuar, como armazenar e transmitir esses dados e quais ações tomar sob variadas circunstâncias.

## 2.5. Código Fonte

Código fonte é o conjunto de palavras e comandos escritos ordenadamente, de maneira lógica, que contém instruções em determinada linguagem de programação.

Há dois tipos de linguagem na hora de programar: as compiladas e as interpretadas.

Compilar uma linguagem de programação é transformar o código fonte em algo que faça sentido para a máquina, ou seja, transformar o que o programador escreveu em linguagem de máquina.

Já no processo de interpretação, o código fonte é transformado numa linguagem intermediária que, por sua vez, será interpretada pela máquina virtual durante a execução do programa.

## 2.6. Habilidades de um programador

É importante também saber que muito além de um simples processo de escrita, a programação envolve um misto de outras habilidades cognitivas, como o raciocínio lógico, habilidades matemáticas, a capacidade de abstração e de lidar com diferentes tipos de dados.

Quando se começa a estudar programação, a primeira coisa que se aprende é criar algoritmos básicos, que são como receitas, para que a partir de um passo a passo, uma tarefa possa ser executada. E esse algoritmo é desenvolvido a partir de tentativas e erros baseados no resultado em que se quer chegar e em como se pode alcançá-lo.

Um exemplo disso é quando se pede a uma pessoa para ir a uma tal padaria, ela vai precisar saber onde ela se localiza para passar as instruções adequadas, como: ande em frente até à rua tal (e também precisará saber o nome da rua) vire à direita (e precisará ter noções de direção) e assim por diante.

Com isso, estará usando a lógica básica para criar um mapa de instruções, assim como os programadores começam a exercitar a lógica para chegar a softwares de computadores, tendo em vista que eles estão instruindo a máquina a chegar a um resultado.

Seguindo no exemplo acima, para chegar a caminhos mais complexos, como construir um prédio, ou criar um software para engenheiros, é preciso passar por esse caminho do básico e ir somando conhecimentos mais complexos, como conhecimentos matemáticos e da própria engenharia, por exemplo.

Por esse motivo é importante dedicar tempo a aprender programação, e quanto mais cedo se começa a dedicar à programação, mais cedo também se alcançam resultados mais complexos, tornando-se um programador cognitivamente capacitado para criar estratégias e soluções tecnológicas muito úteis à nossa sociedade, como softwares médicos, de engenharia, entre outros.

## 2.7. A programação e o avanço das Tecnologias Digitais

A programação teve o seu início há mais de um século, partindo das ideias da Ada Lovelace com o computador robusto do Babbage. Até ao período da segunda Guerra mundial, os hardwares ainda tinham um protagonismo muito maior nos estudos e pesquisas dessas tecnologias do que os softwares, que são o objeto de criação da programação.

Mas, depois da segunda guerra mundial, e principalmente quando os computadores passaram a ganhar uso além do bélico, como os primeiros computadores pessoais e, algumas décadas depois, com a criação dos smartphones, os aparelhos eletrónicos têm solicitado cada vez mais atenção aos softwares, que devem ser tão atualizados quanto os hardwares para que a tecnologia tenha visibilidade e atenda às solicitações, cada vez mais exigentes, do mercado.

Além do uso pessoal, atualmente a programação tem grande importância também em áreas fundamentais à sociedade, como na medicina, onde muitos aparelhos médicos utilizam a monitoração remota e em tempo real, possibilitada pela programação, na engenharia, indústria mecânica, automotiva e arquitetura, onde vários softwares utilizados para a criação de projetos foram desenvolvidos por programadores, entre outras áreas.

Foi o avanço da programação que possibilitou também o que hoje chamamos de Internet das Coisas, que é capaz de automatizar desde pequenos ambientes de uma casa até uma cidade inteira.

Além disso, a programação é uma das principais ferramentas utilizadas no avanço das tecnologias artificiais, que já trazem (a exemplo da Cortana e de invenções como a robô Sophia) muito mais comodidade ao dia a dia, por exemplo ao fornecer informações em tempo real e ser capaz de realizar diversas atividades, como a Sophia, que atua em tratamentos biológicos como o autismo.

## 2.8. O futuro da programação

O novo e mais complexo sentido da palavra “Programação” tem adquirido uma relevância cada vez mais alta nas últimas décadas e, graças ao advento da Indústria 4.0, que é o contexto tecnológico em que estamos inseridos, esse cenário tende a ser permanente e exigir programadores cada vez mais capacitados para atuar numa série de diferentes áreas voltadas para a programação nos próximos anos.

A programação pode ser vista, assim, como uma profissão fundamental no futuro, e é justamente por isso que diversos países se têm esforçado para iniciar o ensino de programação cada vez mais cedo às crianças, para desenvolver nelas habilidades de raciocínio lógico, abstração, entre outras e também para suprir as exigências crescentes da profissão.

## 2.9. Conceito de base de dados

Uma base de dados é uma ferramenta de recolha e organização de informações. As bases de dados podem armazenar informações sobre pessoas, produtos, encomendas ou qualquer outro assunto. Muitas bases de dados começam por ser uma lista num programa de processamento de texto ou numa folha de cálculo. À medida que a lista cresce, começam a aparecer inconsistências e repetições nos dados. Os dados tornam-se difíceis de compreender num formato de lista e as maneiras para efetuar pesquisas ou selecionar subconjuntos de dados para revisão são limitadas. Assim que estes problemas começam a aparecer, convém transferir os dados para uma base de dados criada por um sistema de gestão de bases de dados (DBMS) como o Access.

Uma base de dados informatizada é um contentor de objetos. Uma base de dados pode conter mais do que uma tabela. Por exemplo, um sistema de controlo de inventários que utilize três tabelas não corresponde a três bases de dados mas sim uma base de dados que contém três tabelas. A não ser que tenha sido especificamente concebida para utilização de dados ou código de outra origem, uma base de dados do Access armazena as respetivas tabelas num único ficheiro, em conjunto com outros objetos, tais como formulários, relatórios, macros e módulos. As bases de dados criadas no formato do Access 2007 (que também são usadas no Access 2016, Access 2013 e Access 2010) têm a extensão de ficheiro .accdb e as bases de dados criadas em versões do Access anteriores têm a extensão de ficheiro .mdb. Pode utilizar-se o Access 2016, Access 2013, Access 2010 ou Access 2007 para criar ficheiros em formatos anteriores (por exemplo, Access 2000, Access 2002-2003).

Ao utilizar o Access, pode:

- Adicionar novos dados a uma base de dados, como um novo item num inventário;

- Editar dados existentes na base de dados, como alterar a localização atual de um item;
- Eliminar informações, por exemplo, quando um item é vendido ou retirado;
- Organizar e visualizar os dados de várias formas diferentes;
- Partilhar os dados com outras pessoas através de relatórios, mensagens de e-mail, intranet ou Internet.

Os elementos de uma base de dados do Access são:

- Tabelas
- Formulários
- Relatórios
- Consultas
- Macros
- Módulos

Tabelas



O aspeto de uma base de dados é semelhante a uma folha de cálculo, na qual os dados são armazenados em linhas e colunas. Por esta razão, costuma ser bastante fácil importar os dados de uma folha de cálculo para uma base de dados. A principal diferença entre armazenar os seus dados numa folha de cálculo e armazená-los numa base de dados é a forma como os dados estão organizados.

Para obter a maior flexibilidade de uma base de dados, os dados precisam de ser organizados em tabelas para que não ocorram redundâncias. Por exemplo, se armazenar informações sobre funcionários, cada funcionário apenas deverá ter de ser introduzido uma vez numa tabela que esteja configurada para receber dados de funcionários. Os dados sobre produtos serão armazenados numa tabela específica e os dados sobre sucursais serão armazenados noutra tabela. Este processo é denominado *normalização*.

Cada linha de uma tabela é referida como um registo. Os registos são onde as informações individuais são armazenadas. Cada registo é composto por um ou mais campos. Os campos correspondem às colunas da tabela. Por exemplo, pode ter uma tabela chamada "Empregados" onde cada registo (linha) contém informações sobre um empregado diferente, e cada campo (coluna) contém um tipo diferente de informação, como o primeiro nome, apelido, endereço, e assim por diante. Os campos devem ser designados como um determinado tipo de dados, seja texto, data ou hora, número ou algum outro tipo.

Outra forma de descrever registos é imaginando um catálogo de fichas de biblioteca à moda antiga. Cada ficha no arquivo corresponde a um *registo* na base de dados. Cada informação numa ficha individual (autor, título, etc.) corresponde a um *campo* na base de dados.

## Formulários



Os formulários permitem-lhe criar um interface de utilizador no qual pode introduzir e editar os seus dados. Os formulários contêm frequentemente botões de comando e outros controlos que desempenham várias tarefas. Pode criar-se uma base de dados sem utilizar formulários ao editar os dados nas folhas de dados da tabela. Contudo, a maioria dos utilizadores de bases de dados preferem utilizar os formulários para visualizar, introduzir e editar dados nas tabelas.

Pode programar botões de comando para determinar que dados aparecem no formulário, abrir outros formulários ou relatórios ou desempenhar várias outras tarefas. Por exemplo, pode ter um formulário denominado "Formulário de Clientes" no qual trabalha com dados de clientes. O formulário de clientes pode ter um botão que abre um formulário de encomenda no qual pode efetuar uma nova encomenda para esse cliente.

Os formulários também lhe permitem controlar a forma como os outros utilizadores interagem com a base de dados. Por exemplo, pode criar um formulário que mostre apenas certos campos e permite que apenas certas operações possam ser desempenhadas. Isto ajuda-o a proteger os dados e a assegurar que os dados são introduzidos adequadamente.

## Relatórios



Os relatórios são utilizados para formatar, resumir e apresentar dados. Um relatório, normalmente, responde a uma questão específica, como "Quanto dinheiro recebemos de cada cliente este ano?" ou "Em que cidades estão localizados os nossos clientes?". Cada relatório pode ser formatado para que apresente as informações da forma mais legível possível.

Um relatório pode ser executado a qualquer altura e refletirá sempre os dados atuais na base de dados. Os relatórios são geralmente formatados para poderem ser impressos, mas também podem ser visualizados no ecrã, exportados para outro programa ou enviados como um anexo a uma mensagem de e-mail.

## Consultas



As consultas podem desempenhar várias funções diferentes numa base de dados. A sua função mais comum é a de obtenção de dados específicos das tabelas. Os dados que pretende visualizar estão normalmente distribuídos pelas várias tabelas e as consultas permitem-lhe visualizá-los uma única folha de dados. Para além disso, tendo em conta que o utilizador normalmente não pretende visualizar todos os registos de uma só vez, as consultas permitem-lhe adicionar critérios para "filtrar" os dados de modo a visualizar apenas os registos que pretende.

Algumas consultas são "atualizáveis", o que significa que pode editar os dados nas tabelas subjacentes através da folha de dados da consulta. Caso esteja a trabalhar numa consulta atualizável, lembre-se de que as suas alterações estão de facto a ser efetuadas nas tabelas, não apenas na folha de dados da consulta.

Existem dois tipos de consultas base: consultas de seleção e consultas de ação. Uma consulta de seleção obtém os dados e disponibiliza-os para utilização. Pode visualizar os resultados da consulta no ecrã, imprimi-los ou copiá-los para a área de transferência. Em alternativa, pode utilizar o resultado de uma consulta como a origem do registo para um formulário ou relatório.

Uma consulta de ação, como o nome indica, desempenha uma tarefa com os dados. As consultas de ação podem ser utilizadas para criar novas tabelas, adicionar dados a tabelas existentes, atualizar dados ou eliminar dados.

## Macros



As macros no Access podem ser entendidas como uma linguagem de programação simplificada que pode utilizar para adicionar funcionalidade a uma base de dados. Por exemplo, pode anexar uma macro a um botão de comando num formulário de modo a que a macro seja executada sempre que o botão é clicado. As macros contêm ações que desempenham tarefas, tais como a abertura de um relatório, a execução de uma consulta ou o fecho da base de dados. A maior parte das operações que executa manualmente pode ser automatizada através de macros, o que permite poupar tempo de uma forma extraordinária.

## Módulos



Os módulos, como as macros, são objetos que pode utilizar para adicionar funcionalidades à sua base de dados. Se por um lado o utilizador cria macros no Access

ao escolher a partir de uma lista de ações de macros, os módulos são escritos pelo utilizador na linguagem de programação Visual Basic for Applications (VBA). Um módulo é uma coleção de declarações, instruções e procedimentos que estão armazenados em conjunto como uma unidade. Um módulo pode ser um módulo de classe ou um módulo padrão. Os módulos de classe são anexados a formulários ou relatórios e normalmente contêm procedimentos que são específicos do formulário ou relatório ao qual se encontram anexados. Os módulos padrão contêm procedimentos gerais que não estão associados a qualquer outro objeto. Os módulos padrão são listados em Módulos no Painel de Navegação, ao passo que os módulos de classe não.

## 2.9. Cuidados ao criar uma base de dados

Os descuidos com a base de dados podem afetar o desempenho do site, reduzir a velocidade e até colocar em risco a integridade das informações. É comum ouvir histórias que dentro de um ou dois anos o site passa a ter problemas de desempenho e o motivo é justamente por uma falha na estrutura da base de dados.

Para evitar este tipo de problemas, há cinco aspetos essenciais para uma boa base de dados. Assim:

### 2.9.1. Segmentar as informações

A base de dados pode armazenar diversos tipos de informações, desde nome, e-mail e senha, cartão de cidadão, data de nascimento, endereço, NIF, etc. Para manter tudo organizado, o ideal é criar tabelas separadas para cada tipo de informação, como no exemplo de um e-commerce, teríamos as tabelas abaixo:

- Tabela de Produtos: ficariam as informações dos produtos, como preço, descrição e etc.;

- Tabela de Clientes: onde ficam, por exemplo, os dados do consumidor e endereço de entrega;
- Tabela de Vendas efetuadas: que seria a relação entre as duas informações acima.

### 2.9.2. Definir os índices e relações

Seguindo o conceito de modelagem, definir o que são informações únicas, que não devem ser repetidas, como o Cartão de Cidadão, por exemplo. Caso alguém tente registar-se usando um CC já registado o sistema não aceita, pois é um índice único.

É necessário estabelecer a relação entre as tabelas, onde a chave estrangeira é criada para fazer a ligação dos dados.

### 2.9.3. Preservar a cronologia das informações

É relevante guardar a data em que um dado foi adicionado ao sistema ou um novo registo foi criado. A base de dados pode fazer esse registo automaticamente, através da sua propriedade de valor padrão. Essa informação é muito útil no momento de resolver problemas.

### 2.9.4. Priorizar a integridade dos dados

Deve ter-se em atenção a segurança das informações que estão na base de dados. O básico é definir quem terá acesso direto à base de dados, ao servidor e uma política de segurança, que inclua backups regulares (preferencialmente diários).

### 2.9.5. Definir os tipos de dados

Definir quais são as propriedades de cada informação armazenada, como por exemplo os campos de NIF e telefone, necessariamente, precisam ser caracteres numéricos. E também definir limitações de tamanho dos dados, como um nome não deve ultrapassar 200 caracteres.

É importante detetar falhas numa base de dados. Muitas vezes são coisas básicas, como uma data que o sistema regista como texto ou detalhes mais avançados como uma requisição na base de dados mal construída.

E, para quem quer aperfeiçoar e melhorar o seu desempenho profissional, a orientação é adquirir conhecimento, praticar e dedicar-se ao máximo.

## 2.10. História das Bases de Dados

Desde os primórdios da informática que existem duas operações que andam de mãos dadas: guardar informação e obter essa mesma informação de volta. Estas operações, por mais simples que pareçam, têm as suas dificuldades e são elas que vão determinar em grande parte a fiabilidade e a eficiência de todo o sistema envolvente.

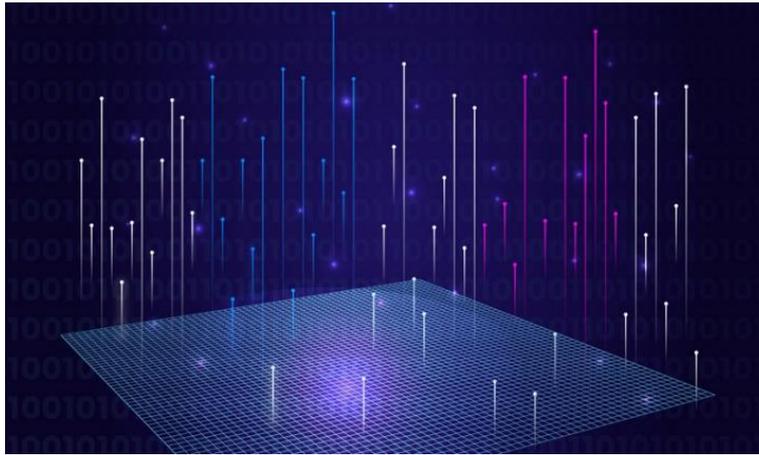


Figura 2 – Processo de guardar e obter dados

Os processos de guardar e de obter dados são dos mais importantes no software de hoje em dia. Por isso, para minimizar o tempo de resposta do sistema aquando destas operações, é preciso que as formas com que acedemos à memória sejam inteligentes e eficazes.

Inicialmente, a única solução que os programadores tinham era guardar os dados das suas aplicações em ficheiros. Esta situação levanta uma série de questões:

- é complicado manipular ficheiros, principalmente quando é necessário lidar com vários em simultâneo;
- a eficiência desta lógica era baixa, por exemplo, para ir buscar um cliente, teria que se carregar todos os clientes para memória e só depois aplicar o filtro (e talvez fosse necessário carregar outros ficheiros dependendo da organização da informação);

- se um registo fosse mal guardado poderia comprometer todo o ficheiro e os que com ele se relacionam;
- não há abstração nenhuma entre a manipulação dos dados e a maneira como eles foram guardados.

Para solucionar estes problemas, por volta dos anos 60, surgiram os primeiros sistemas de gestão de bases de dados (SGBD) – um conjunto de ficheiros organizados que tinha como principal objetivo retirar da aplicação cliente a responsabilidade de gerir o acesso, a persistência, a manipulação e a organização dos dados.



Figura 3 - SGBD

Estes SGBD seguiam o modelo de navegação – modelo procedimental que permitia ao programador escolher um caminho arbitrário seguindo as relações de registo para registo, idêntico a um grafo ou uma rede.

Facto curioso é que este modelo apenas foi possível com a criação do disco magnético, pois este permitia um acesso random aos dados armazenados, ou seja, o programador podia saltar entre zonas não contíguas do disco, ao contrário dos seus antecessores (como a fita e cartões perfurados) que obrigavam a ler os dados armazenados de forma sequencial.

Charles Bachman foi um dos grandes impulsionadores deste modelo, o qual usou para construir um dos primeiros SGBD, o IDS (Integrated Data Store).



Figura 4 - Charles Bachman

Apesar do modelo de navegação resolver muitos dos problemas referidos, continuava a haver um sério problema por resolver. Caso a estrutura interna dos dados fosse mudada, os programas que usavam esses mesmos dados tinham que ser todos alterados e reestruturados, pois continuava a não haver a abstração entre a aplicação cliente e os dados.

Em 1970, enquanto trabalhava para a IBM, Edgar Frank Codd, descontente com esta situação, publicou o seu artigo “A Relational Model of Data for Large Shared Data Banks” onde descreveu o conceito de modelo relacional – modelo baseado na teoria de conjuntos em que uma base de dados é vista como uma coleção de relações (nos SGBDs são as tabelas), que, por sua vez, são conjuntos de tuplos (linhas nos SGBDs), que são grupos de atributos (colunas nos SGBDs).

Este modelo permite, através das suas características, que os dados sejam completamente desacoplados da aplicação, visto que já não se acede aos dados de uma maneira procedimental.

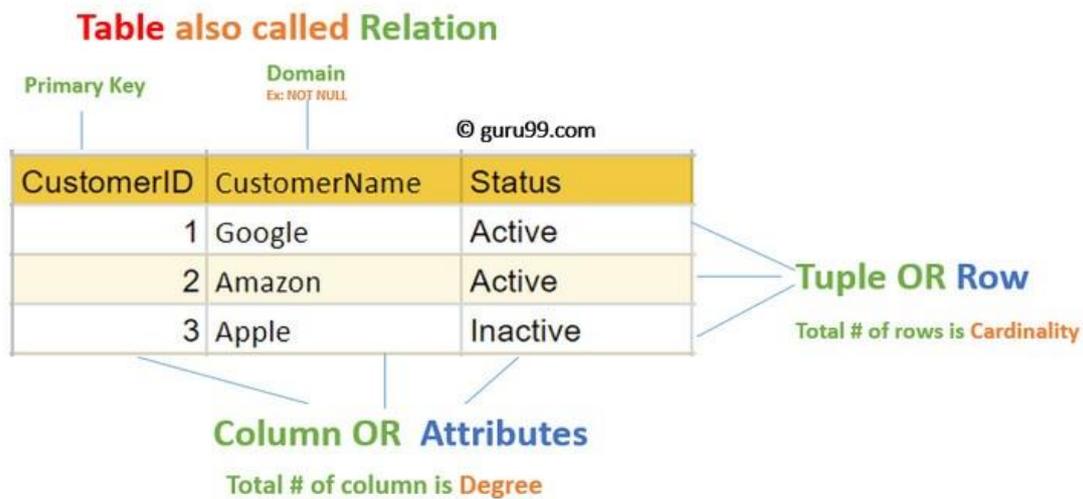


Figura 5 – Modelo Relacional

Em 1973, dois cientistas da Universidade de Berkeley, Michael Stonebraker e Eugene Wong, leram o trabalho publicado por Codd e ficaram muito interessados na ideia do modelo relacional.

Nesta altura, estes dois cientistas estavam na posse de fundos que tinham angariado para a pesquisa de um sistema de base de dados geográfica para o grupo de economia da Universidade de Berkeley, o Ingres (INteractive Graphics RETrieval System). Mas, visto o seu interesse e entusiasmo com o modelo relacional, Stonebraker e Wong decidiram usar esse dinheiro não para o seu destino original, mas sim para um SGBD relacional ao qual deram o mesmo nome do sistema de base de dados que era suposto ser feito à partida.

Ao mesmo tempo, desenvolveram a linguagem de consulta que iria ser usada para pesquisar e alterar a base de dados, a QUEL.

Entretanto, dois jovens que na altura, tal como Edgar F. Codd, trabalhavam na IBM, Donald D. Chamberlin e Raymond F. Boyce, inspirados pelo modelo relacional, desenvolveram uma linguagem declarativa idêntica ao QUEL, mas com mais funcionalidades, com o objetivo de manipular e de obter dados guardados do System R, a base de dados relacional que a IBM estava a desenvolver na altura. Essa linguagem

era o SEQUEL – Structured English Query Language ou, em português, Linguagem Estruturada de Consulta em Inglês, mais tarde abreviado para SQL.



Figura 6 -SQL

Estes dois SGBDs demonstraram que bases de dados relacionais podiam ser eficazes e eficientes, validando assim o modelo relacional e revolucionando o mundo informático com este novo paradigma.

- Resumo (e algumas curiosidades):

- **1963:** Charles Bachman lança o IDS, um dos primeiros SGBDs do mundo;
- **1970:** Edgar Frank Codd publica o seu artigo onde explica o seu conceito de modelo relacional;
- **1973:** Charles Bachman ganha o Prémio de Turing pelo seu modelo de navegação e publica, devido a este acontecimento, "*The Programmer as Navigator*";

– Michael Stonebraker e Eugene Wong começam a trabalhar no Ingres e no QUEL;

- **1974:** o System R começa a ser desenvolvido e o SQL é desenhado por Donald D. Chamberlin e Raymond F. Boyce.

– Lançado um primeiro protótipo do Ingres

- **1977:** System R tem o seu primeiro cliente, Pratt & Whitney;
- **1981:** Edgar F. Codd ganha o Prémio de Turing devido ao seu modelo relacional, publicando *“1981 Turing Award Lecture – Relational Database: A Practical Foundation for Productivity”*;
- **2014:** Graças às suas fundamentais contribuições para os conceitos e práticas subjacentes aos modernos SGBDs, Michael Stonebraker recebe o Prémio de Turing.

O SQL, usado ainda hoje em bases de dados de alta performance, é provavelmente das linguagens mais importantes na história da computação. No entanto, o SQL é apenas uma linguagem e precisa de programas que explorem ao máximo as suas potencialidades, tal como o System R. Estes programas são os já referidos SGBD. Por isso, devido à importância que estes têm, no próximo artigo de “A História das Bases de Dados” serão abordados os principais e mais marcantes SGBD relacionais e a sua história.

## Capítulo III – Programas para criar sites

### 3.1. Dreamweaver

O Dreamweaver é um programa de edição de páginas da internet, onde se pode trabalhar com imagens, textos, e muitos outros elementos para a Web.

O programa permite a criação e edição de páginas na internet, sendo em alguns aspectos, muito parecido com programas de criação de layouts. Com este programa não é necessário criar um código de programação, acelerando assim o tempo de produção de sites, embora seja muito complicado trabalhar com este programa sem conhecimentos básicos de layout de sites. O Dreamweaver tem suporte à maioria das linguagens estruturadas do mercado. O programa cria códigos na linguagem desejada, e o utilizador só precisa utilizar a interface gráfica. O utilizador pode colocar uma imagem no seu site, e o programa escreve o código para a página da Web.

O programa era originalmente da Macromedia, sendo adquirido pela Adobe, que lançou sua primeira versão do programa no mercado em 2007, o Dreamweaver CS3. A partir de então, a Adobe vem investindo muito no programa, que já esta na versão CS5. Esta nova versão trouxe muitas melhorias para o programa. O Dreamweaver é um dos melhores programas da área, sendo o mais usado pelos profissionais.

A Adobe comprou o Dreamweaver em uma época de expansão da empresa. Os programas que tem CS (Adobe Creative Suite) no nome fazem parte de um pacote conhecido como pacote Adobe. Este pacote tem programas, sendo que todos eles têm alguma aplicação em design gráfico, desenvolvimento web ou edição de vídeo. Entre estes programas, o Dreamweaver é um dos mais importantes atualmente.

Um dos problemas com o Dreamweaver é que algumas funções criam os chamados “códigos sujos”. Estes são códigos com linhas em excesso, que no inicio não atrapalham o site. Porém, se um programador for olhar o código, para editar algo, ou corrigir algum problema, estes códigos sujos atrapalham, dificultando a edição.

Este programa é muito útil para os criadores de sites, porém é sempre bom a pessoa aprender as linguagens de programação, para fazerem seus próprios códigos, assim o Dreamweaver torna-se um programa para consultar, e lembrar de alguns códigos, durante a programação.

Algumas linguagens mais usadas no Dreamweaver, na versão CS5:

ActionScript

Active Server Pages (ASP).

ASP.NET

C#

Cascading Style Sheets (CSS)

ColdFusion

EDML

Extensible HyperText Markup Language (XHTML)

Extensible Markup Language (XML)

Extensible Stylesheet Language Transformations (XSLT)

HyperText Markup Language (HTML)

Java

JavaScript

JavaServer Pages (JSP)

PHP: Hypertext Preprocessor (PHP)

### 3.2. Notepad++

Esta é outra aplicação muito usada em sala de aula, nas aulas de Sistemas de Informação. É uma aplicação gratuita para fazer o download e de fácil aprendizagem em termos de layout da própria aplicação.

Notepad++ é um editor de texto e de código fonte de código aberto sob a licença GPL. Suporta várias linguagens de programação rodando sob o sistema Microsoft Windows (possível utilização no Linux via Wine).

O Notepad++ é distribuído como um software livre. O projeto foi hospedado no SourceForge.net, onde foi baixado mais de 27 milhões de vezes e ganhou duas vezes o premio "SourceForge Community Choice Award" como melhor ferramenta de desenvolvimento. O projeto foi hospedado no TuxFamily desde 2010 até 2015 depois foi movido para o GitHub. Ele é baseado no Scintilla, é escrito em C++ utilizando a API Win32 e usa a STL. O objetivo do Notepad++ é oferecer um estreito e eficiente binário com uma interface gráfica totalmente modificável.

As linguagens de programação suportadas pelo Notepad++ são: C, C++, Java, C#, XML, HTML, PHP, JavaScript, makefile, ASCII art, doxygen, ASP, VB/VBScript, Unix Shell Script, BAT, SQL, Objective-C, CSS, Pascal, Perl, Python, Lua, Tcl, Assembly, Ruby, Lisp, Scheme, Smalltalk, PostScript e VHDL. Além disto, os utilizadores podem definir as suas próprias linguagens usando um "sistema de definição de linguagem" integrado, que faz do Notepad++ extensível, para ter realce de sintaxe e compactação de trechos de código.

Ele suporta Auto complemento, busca e substituição com integração de expressões regulares, divisão de tela, zoom, favoritos, etc. Tem suporte para macros e plugins. Um plugin de utilizador chamado TextFX, que permite opções de transformação de textos, é incluído por padrão.

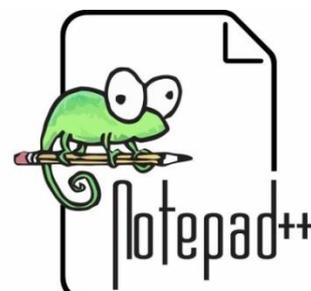


Figura 7 – Logotipo do Notepad++

### 3.3. Python

Python é uma linguagem de programação de alto nível, interpretada de script, imperativa, orientada a objetos, funcional, de tipagem dinâmica e forte. Foi lançada por Guido van Rossum em 1991. Atualmente, possui um modelo de desenvolvimento comunitário, aberto e gerido pela organização sem fins lucrativos Python Software Foundation. Apesar de várias partes da linguagem possuírem padrões e especificações formais, a linguagem, como um todo, não é formalmente especificada.

A linguagem foi projetada com a filosofia de enfatizar a importância do esforço do programador sobre o esforço computacional. Prioriza a legibilidade do código sobre a velocidade ou expressividade. Combina uma sintaxe concisa e clara com os recursos poderosos de sua biblioteca padrão e por módulos e frameworks desenvolvidos por terceiros.



Figura 8 – Logotipo do Python

Python é uma linguagem de propósito geral de alto nível, multiparadigma, suporta o paradigma orientado a objetos, imperativo, funcional e procedural. Possui tipagem dinâmica e uma de suas principais características é permitir a fácil leitura do código e exigir poucas linhas de código e compara-se ao mesmo programa noutras linguagens. Devido às suas características, é utilizado, principalmente, para processamento de textos, dados científicos e criação de CGIs para páginas dinâmicas para a web. Foi considerada pelo público a 3ª linguagem "mais amada", de acordo com uma pesquisa conduzida pelo site Stack Overflow em 2018 e está entre as 5 linguagens mais populares, de acordo com uma pesquisa conduzida pela RedMonk.

O nome Python teve a sua origem no programa Monty Python's Flying Circus, teve origem no grupo humorístico britânico Monty Python, embora muitas pessoas façam associação com o réptil do mesmo nome (em português, píton ou pitão).

## Capítulo IV – Desenvolvimento do Projeto

## 4.1. Aplicação utilizada

"IntelliJ" foi utilizado na maior parte para a preparação da base de dados.

O Dreamweaver foi utilizado mais para embelezamento do site que os utilizadores vão visualizar.

## 4.2. Barras de Ferramentas e Algumas Funcionalidades

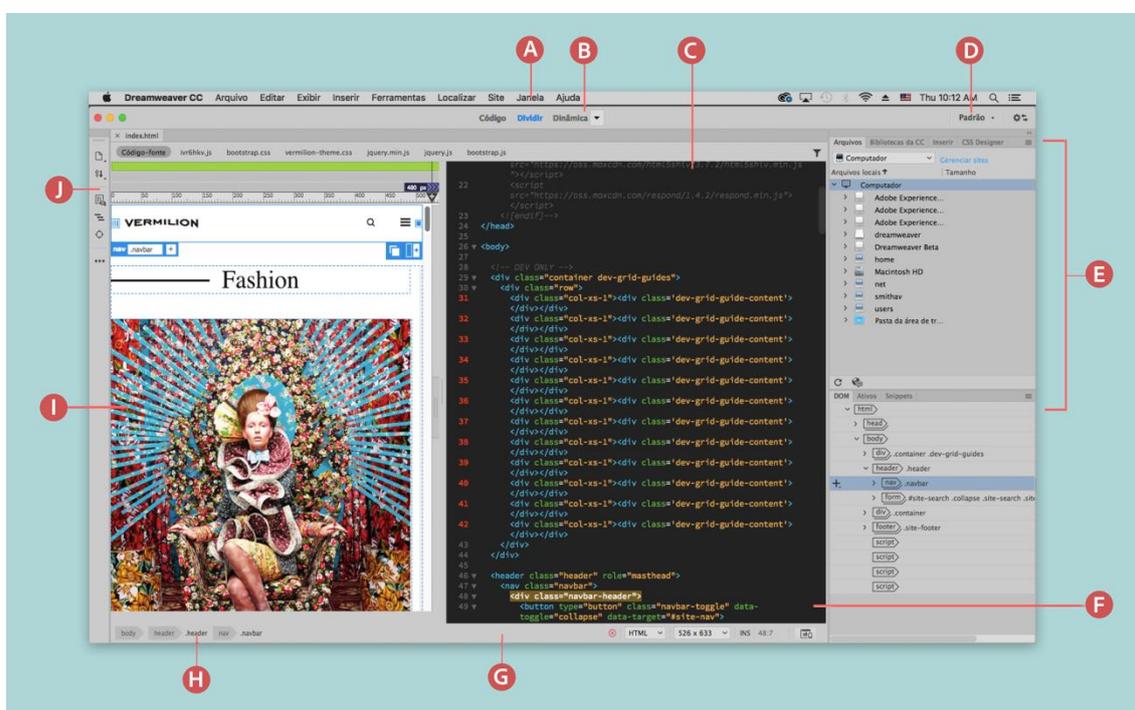


Figura 9 - Ambiente de trabalho do IntelliJ

A área de trabalho inclui os seguintes elementos:

- A.** Barra de aplicativos
- B.** Barra de ferramentas Documento
- C.** Janela do documento
- D.** Alternador da área de trabalho
- E.** Painéis
- F.** Visualização de código
- G.** Barra Status
- H.** Seletor de tags
- I.** Visualização dinâmica
- J.** Barra de ferramentas.

## A barra de aplicativos

Localizada na parte superior da janela do aplicativo, contém o alternador da área de trabalho, os menus (apenas no Windows) e outros controles do aplicativo.

## A barra de ferramentas Documento

Contém botões que fornecem opções para diferentes visualizações da janela Documento (como a Visualização de design, dinâmica e de código).

## A barra de ferramentas Padrão

Para exibir a barra de ferramentas Padrão, selecione Janela > Barras de ferramentas > Padrão. A barra de ferramentas contém botões para operações comuns dos menus Arquivo e Editar: Novo, Abrir, Salvar, Salvar tudo, Imprimir código, Cortar, Copiar, Colar, Desfazer e Refazer.

## A barra de ferramentas

Está à esquerda da janela do aplicativo e contém botões específicos a visualizações.

## A janela do documento

Exibe o documento atual à medida que é criado e editado.

## O Inspetor de propriedades

Permite exibir e alterar diversas propriedades do objeto ou texto selecionado. Cada objeto tem propriedades diferentes.

## O seletor de tags

Localizado na barra Status na parte inferior da janela Documento. Mostra a hierarquia de tags em torno da seleção atual. Clicar em qualquer tag da hierarquia para selecionar a tag e todo seu conteúdo.

## Painéis

Ajuda a monitorar e modificar seu trabalho. Os exemplos incluem o painel Inserir, o painel CSS Designer e o painel Arquivos. Para expandir um painel, clicar duas vezes na guia correspondente.

## O painel Extract

Permite fazer o upload e exibir os arquivos PSD na Creative Cloud. Usando esse painel, pode extrair CSS, texto, imagens, fontes, cores, gradientes e medidas das composições de PSD no documento.

## O painel Inserir

Contém botões para inserir vários tipos de objetos em um documento, como imagens, tabelas e elementos de media. Cada objeto é uma parte do código HTML que permite definir vários atributos à medida que são inseridos. Por exemplo, pode-se inserir uma tabela clicando no botão Tabela no painel Inserir. No entanto, pode-se inserir objetos usando o menu Inserir em vez do painel Inserir.

## O painel Arquivos

Permite gerir arquivos e pastas, sejam eles parte de um site do Dreamweaver ou de um servidor remoto. Com o painel Arquivos também se tem acesso a todos os arquivos no disco local. Para obter mais informações, consultar Gerenciamento de arquivos e pastas.

## O painel Snippets

Permite guardar e reutilizar os snippets de código em diferentes páginas da Web, sites e instalações do Dreamweaver (usando a sincronização de configurações). Para obter mais informações, consultar Reutilizar códigos com snippets.

## O painel CSS Designer

É um Inspetor de propriedades CSS que permite criar “visualmente” estilos e arquivos CSS e definir propriedades juntamente com consultas de media.

## Capítulo V – Conclusões

## 5.1. Análise Crítica

Começo por referir que todos os prazos e objetivos foram alcançados o que me faz gerar um sentimento de satisfação no que se refere ao trabalho realizado.

Ao longo dos três anos de curso aprendi muita coisa, óbvio que tive dificuldades, mas que superei com a ajuda dos meus professores e colegas, tudo isto conectado à minha motivação e ao facto de nunca desistir quando um obstáculo se aproximava.

Destaco também as muitas horas de trabalho investido ao longo do ano através de um trabalho contínuo e progressivo.

## 5.2. Conclusão

A aquisição de novas competências foi uma constante ao longo destes 3 anos, que culminou na elaboração desta PAP. Além de colocar em prática muitos conhecimentos abordados ao longo do curso, trouxe-me a possibilidade de adquirir novas competências na área da programação.

Não posso também deixar de expressar o meu agradecimento a todos os intervenientes que diretamente ou indiretamente estiveram envolvidos neste projeto, destacando o forte apoio dos meus colegas de turma e de todos os professores que partilharam os seus conhecimentos ao longo destes três anos.

Para concluir gostei muito de realizar esta PAP principalmente porque foi uma ótima experiência de aprendizagem, podendo ser uma mais-valia para um futuro no mercado de trabalho.

## Webgrafia

- <https://helpx.adobe.com/br/dreamweaver/using/dreamweaver-workflow-workspace.html>
- [https://en.wikipedia.org/wiki/Adobe\\_Dreamweaver](https://en.wikipedia.org/wiki/Adobe_Dreamweaver)
- <https://www.infoescola.com/informatica/dreamweaver/>
- <https://pplware.sapo.pt/software/top-5-ferramentas-para-gerir-bases-de-dados/>
- <https://conceito.de/registo-de-dados>