**Escola Secundária de Silves**

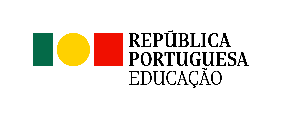
**Curso Profissional de Técnico de Informática - sistemas**

**<Título do Relatório da PAP>**

[Figura Apropriada]

<Nº 12285> <Ricardo Leandro Medina Cabrita>

**<Data>**



**Escola Secundária de Silves**

**Curso Profissional de Técnico de Informática - Sistemas**

**<data>**

**<Título do Relatório da PAP>**

**Relatório Final**

**da**

**Prova de Aptidão Profissional**

<Nº 12285> <Ricardo Cabrita>

Orientado pelos Professores: <Nome dos Professores>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Nome do Aluno) (Professores responsáveis)

Dedicatória [Se existir]

Esta secção apresenta a dedicatória à família, pai, mãe, namorada/0, ou alguma outra pessoa ou organização a quem o aluno deseje dedicar o trabalho.

Normalmente não existe qualquer título e o texto é alinhado à direita.

**Agradecimentos** [Se existir]

Nesta secção colocam-se notas de agradecimento às pessoas e entidades que de alguma forma contribuíram para a realização do trabalho: família, professores, pessoas com quem discutiu ideias, pessoas que forneceram dados/informação ou ajudaram de alguma maneira, escola, ou outras entidades.

Os agradecimentos devem ser diretos e específicos, explicitando o contributo de cada uma das pessoas a quem se agradece.

O título não deve ser numerado.

**Resumo**

O *resumo* do relatório é uma representação abreviada e precisa, sem acrescento de interpretação ou crítica, escrita de forma impessoal, podendo ter, por exemplo, as seguintes três componentes:

1. um parágrafo inicial de introdução do contexto geral do trabalho.
2. resumo dos aspetos mais importantes do trabalho desenvolvido.

Deve mencionar tudo aquilo que foi feito, por isso deve concentrar-se no que é realmente importante e que deve ajudar o leitor a decidir se deve ou não consultar o restante relatório.

1. um parágrafo final com as conclusões do trabalho realizado.

**[Secção Opcional]**

**Palavras Chave (Tema):** incluir 3 a 5 palavras chave que caracterizem o projeto do ponto de vista de tema/área de intervenção.

**Palavras Chave (Tecnologias):** incluir 3 a 5 palavras chave que caracterizem o projeto do ponto de vista de tecnologias utilizadas.

A ordem pela qual se escrevem deve ser decrescente relativamente à sua importância.

O Resumo deve ocupar no máximo uma página.

Os títulos Resumo e Palavras-chave não devem ser numerados.

Dica

Só deve ser escrito após o texto principal do relatório estar completo.

Exemplo de um índice:

**Índice**

[Introdução 2](#_Toc156927998)

[Contexto 2](#_Toc156927999)

[Descrição técnica 2](#_Toc156928000)

[Análise do Sistema 2](#_Toc156928001)

[Desenvolvimento 2](#_Toc156928002)

[Avaliação 2](#_Toc156928003)

[Conclusão 2](#_Toc156928004)

[Bibliografia 2](#_Toc156928005)

[Anexo A - Listagem do material utilizado 2](#_Toc156928006)

[Anexo B - Software 2](#_Toc156928007)

Outros Índices

Se o número de figuras ou tabelas não for reduzido, é conveniente elaborar índices individuais para esses elementos.

É também conveniente apresentar listas de abreviaturas, símbolos e temos técnicos, sempre que estes existam em grande número.

Exemplos de outros índices:

**Índice de Figuras**

[*Figura 1 - Exemplo de imagens a) difícil leitura; b) fácil leitura 17*](#_heading=h.1ksv4uv)

[*Figura 2- exemplo de lista de bibliografia 20*](#_heading=h.44sinio)

**Índice de Tabelas**

[*Tabela 1 - estilos pré-definidos 16*](#_heading=h.2jxsxqh)

[*Tabela 2 - Exemplo de tabela 18*](#_heading=h.z337ya)

**Notação** e **Glossário**

|  |  |
| --- | --- |
| **WWW** | World Wide Web |
|  |  |
|  |  |
|  |  |

«Esta secção apresenta os conceitos (glossário de termos) ordenados alfabeticamente, símbolos matemáticos e acrónimos **utilizados** no corpo do texto do relatório.»

# Introdução

Primeiro Capítulo

No capítulo Introdução pretende-se fornecer ao leitor informação suficiente para que este possa compreender os objetivos e o âmbito do trabalho desenvolvido.

Seções deste capítulo:

* **Enquadramento e motivação:**

Identificar a área em que se enquadra o trabalho e o que levou o aluno a desenvolver o trabalho;

* **Descrição do tema:**

Descrever detalhadamente o problema que se pretende resolver com o projeto.

* **Objetivos**:

Os detalhes do problema são convertidos em objetivos concretos de pesquisa e desenvolvimento. Um objetivo consiste num fim concreto que se pretende atingir.

* **Estrutura do relatório:**

Descrever sumariamente o conteúdo dos restantes capítulos do relatório.

O capítulo Introdução deverá dar resposta às seguintes questões:

Qual é o tema?

Porque é importante?

Quais são os objetivos que se pretende alcançar?

De que maneira se pretende alcançar esses objetivos?

Como está organizado o resto do relatório?

Outros Capítulos

Para além dos Capítulos Introdução e Conclusão, os restantes capítulos constituem a descrição detalhada do trabalho desenvolvido pelo aluno.

A estruturação desta descrição dependerá da linha de raciocínio seguida e as atividades realizadas pelo aluno, mas genericamente podemos considerar a seguinte estrutura (capítulos):

# Contexto

Esta seção destina-se a efetuar uma apresentação do projeto, enquadrando-o num contexto mais global, e descreve todos os aspetos relacionados com o trabalho, nomeadamente **as diversas tecnologias utilizadas no seu desenvolvimento**.

Dependendo de cada projeto específico, este conteúdo pode ser uma seção da introdução ou ser colocado em capítulo separado logo após a introdução. O aluno deve discutir com o professor orientador qual a melhor abordagem para o seu caso.

Neste capítulo também se podem apresentar as tecnologias ou metodologias utilizadas no desenvolvimento. Caso as tecnologias tenham sido escolhidas no âmbito do trabalho, deve ser apresentada uma justificação para essa decisão. Não devem, no entanto, apresentar secções exaustivas sobre as tecnologias a menos que se tratem de assuntos recentes. Exemplos de tecnologias que não é necessário apresentar de forma exaustiva são:

* Tecnologias web (ex., PHP, ASP, ASP.net, JSP)
* Sistemas Gestores de Bases de dados (ex., Oracle, SQL Server, MySQL, PHPMyAdmin)
* Plataforma Java ou .NET
* Servidores web (ex., Apache e IIS)
* Linguagens de programação e ambientes de desenvolvimento

Para as tecnologias anteriores bastará uma pequena secção referindo a sua utilização.

No caso de tecnologias recentes devem apresentar uma secção (ou capítulo separado) que forneça o mínimo de conhecimento ao leitor do relatório para compreensão do mesmo.

Descrição dos detalhes técnicos e conceitos que permitem a um indivíduo leigo na matéria a compreensão da abordagem adotada na resolução do(s) problema(s).

Descrição sumária de soluções já existentes.

# Descrição técnica

Para o desenvolvimento da aplicação em questão utilizei um paradigma de desenvolvimento OOP-programação orientada a objetos, construindo uma framework tipoMVC

**O que é MVC?**

M -> Modelo: manipula as conexões do banco de dados e passa informações para o controlador.

V -> View: este manipulador das visualizações do usuário, como templates no front-end.

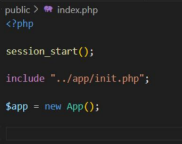
C-> Controlador: este manipulador controla toda a lógica negociável entre todas as partes, o Modelo e as Visualizações.

Primeiro comecei por instalar algumas ferramentas de programação :

* Visual Studio Code;
* Xammp;

De seguida criei uma pasta com o nome “PAP” dentro dessa pasta criei o meu projeto com o nome “papshop” dentro do projeto criei duas pastas “app” (backend) e “public”(frontend)

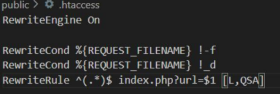
Dentro da pasta publica criei mais dois arquivos “índex.php”,”.htaccess”.



Dentro do “índex.php” inseri o código como esta na imagem

Este código inicia a sessão em nosso navegador e incluímos

Na pasta “app” para instanciar um objeto da nova função APP().

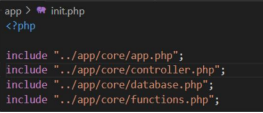
Dentro do arquivo “.htaccess” inseri o seguinte codigo:

Este conjunto de regras do manipulador direciona todos os URLs

solicitações que não correspondem a um arquivo ou diretório

existente ao arquivo index.php, permitindo que o script PHP

processe a URL e decida como lidar com a solicitação.

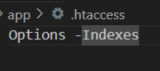
Dentro da parta “APP” criei 2 arquivos “init.php”,”.htaccess” e quatro pastas “controllers”,”core”,”models”,”views”.

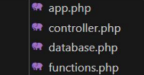
Dentro do arquivo “init.php” dentro

deste arquivo estão todos os caminhos

para as principais estruturas

como esta na imagem

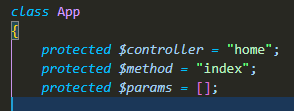
dentro do arquivo “.htaccess” a opção Options -Indexes estiver presente, o servidor web não mostrará uma lista de arquivos quando alguém acessar um diretório no navegador e não houver nenhum arquivo de índice presente. Esta configuração é uma medida de segurança comum para impedir que visitantes vejam a estrutura de diretórios do site e tenham acesso aos arquivos que não deveriam ser públicos.



Na pasta core inseri os seguntes arquivos



E na pasta controller criei os seguintes arquivos

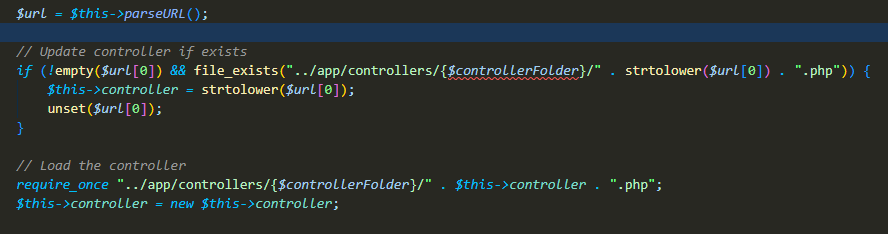
Dentro do arquivo “app.php” coloquei este código

A class App é o ponto de entrada da aplicação

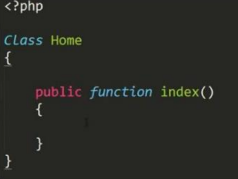
Ele define as propriedades para o “$controller”

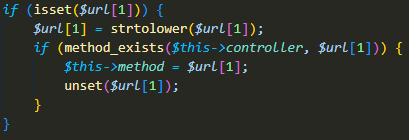
“$method” e os parâmetros da URL “$params”

De seguida eu escrevi o seguinte código :



Em seguida adicionei na pasta “controllers” um novo arquivo “home.php” que inseri o seguinte código:

Dentro da pasta “public” criei dois ficheiros o “índex.php”



Eu criei um código para verificar se o

método existe e, se existir, ler o método,

caso não exista, enviará o usuário para uma

página de erro. Para isso, é criado um

sistema que verifica se a URL contém um

segundo segmento, que geralmente é

usado para especificar um método dentro

do controlador. Se o método existir no

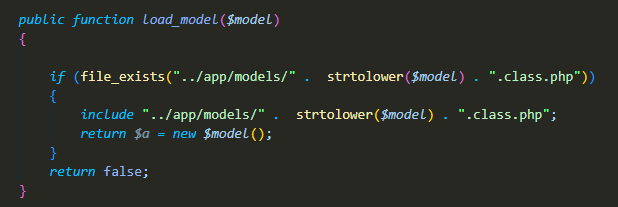
controlador, ele será definido como o método

atual e o segmento correspondente da URL

será removido. Isso permite chamar métodos

específicos dentro de controladores com base na URL

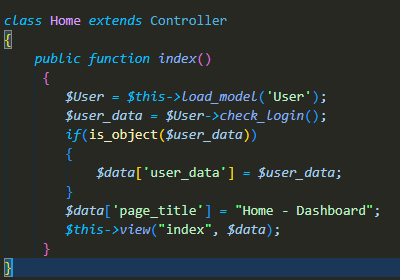
Dentro do arquivo “controller.php” inseri o seguinte código:



Este código recebe o caminho da view verifica a existência e carrega a view que é exigida para o utilizador ver.

em seguida no arquivo “home.php” localizado na pasta “user” fiz uma ligação OPP para extender o controlador inicial para este controlador,porque a função de índice precisa acessar a função de visualização do controlador para poder exibir as visualizações para o utilizador.

Criei um arquivo dentro da pasta “views” que dei o nome de “índex.php” esse arquivo é o que carrega a pagina principal da aplicação

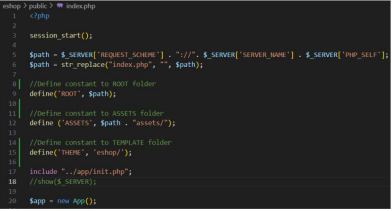
Em seguida inseri o resto do código no arquivo “home.php” :

Este código é o que controla a view da principal da aplicação para que mostre na aplicação.

Um pasta dentro das views “\_includes” dentro dessa pasta contem três arquivos “user\_footer.php”,”user\_header.php”,”user\_menu.php”,fiz isto para organizar e otimizar melhor a minha aplicação que neste caso estes arquivos servem para se alterar alguma coisa dentro deles em toda a aplicação sera alterada ao mesmo tempo sem ter que ir a cada view separadamente fazer as alterações.



A ligação é feita por essa linha de código mas é diferente em baixo e em cima em baixo é alterado de “user\_menu” para “user\_footer” assim ligação é feita com o devido arquivo.

Dentro do arquivo “índex.php” localizado na pasta “public” inseri o seguinte codigo

Este código constrói o caminho base da aplicação, define constantes para os diretórios ,e cria uma instancia da classe ‘app’ para iniciar a aplicação.

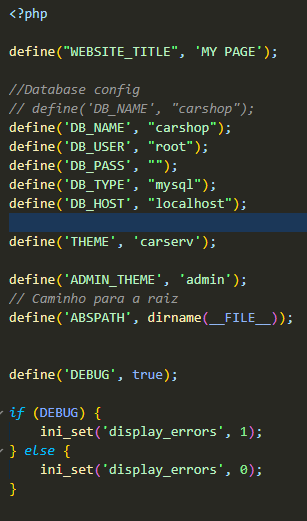
Para fazer as imagens aparecerem na aplicação substitui os caminhos defenindo uma rota direta igual ao seguinte código:



Assim fica mais fácil para lidar com a localização das imagens.

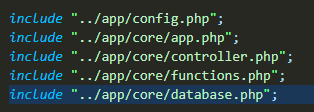
Criei dois arquivos na pasta controllers “register.php”,”login.php”

Criei um arquivo na pasta “app” com o nome “config.php” e inseri o seguinte código:

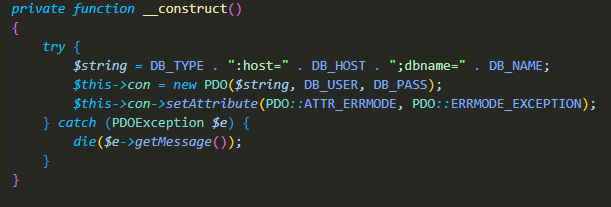


Este código faz a identificação e a conexão com a base de dados.

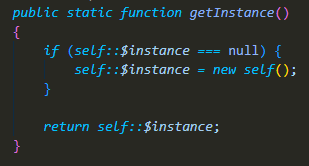
No arquivo “init.php” fiz a referencia de todos os arquivos que fazem a aplicação funcionar e o código que inseri é este:



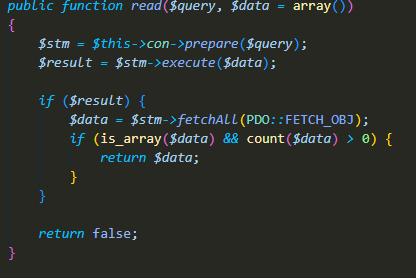
Criei a minha base de dados “carshop” para a aplicação para poder guardar os dados dos utilizadores e dos produtos que serão adicionados mais tarde.

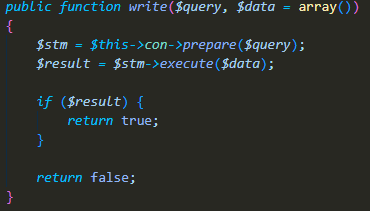
No arquivo “database.php” localizado na pasta “core” adicionei o seguinte código para fazer a ligação via PDO: 

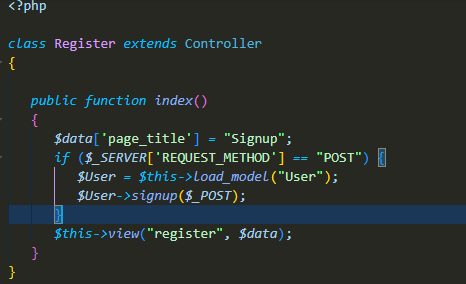
E para garantir apenas exista uma instancia do objeto de conexão exista durante todo o tempo de utilização da aplicção implementei este codigo logo a baixo da ligação PDO:



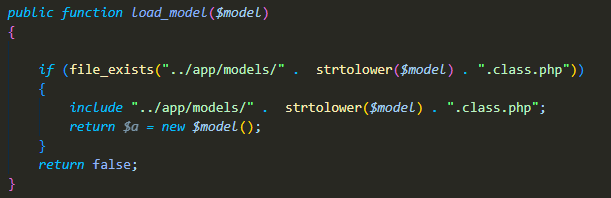
Restrigindo a criação de novas instancias e permitindo apenas uma evita a subrecarga de multiplas conexões na base de dados assim permitindo um bom desempenho da base de dados e do servidor da aplicação.

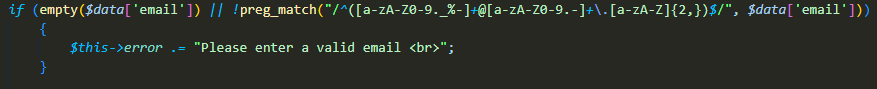
Tambem criei outra função para ler os dados da base de dados fazendo as consultas pela base de dados e recuperar dados o codigo que utilizei é foi o que demosntra na imagem: 

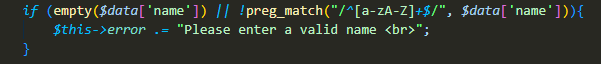
e por fim no arquivo “database.php” inseri o codigo para escrever na base de dados para que possa guardar os dados que eu desejar ou por exemplo quando um utilizador novo criar uma conta os dados serem guarados na base de dados eo codigo que utilizei foi o seguinte: 

dois arquivos na pasta controllers uma para o login eoutra para o register,dentro do “register.php” inseri o seguinte codigo que recebe um “$POST” quando o utilizador clica em registrar e cria um documento json e manda para a base de dados: 

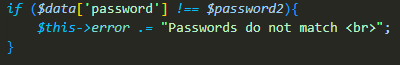
inseri tambem mais uma função no arquivo “controller.php” localizado no “core” que recupera o arquivo de “user.class.php” da pasta models para enviar os dados para a base de dados:

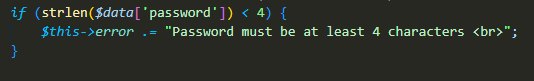


O arquivo “user.class.php” como mencionei no codigoanterior é usado para atribuir funções exemplo :validar um email como vou mostrar no codigo a função validação de email: 

Tambem a validação do nome: 

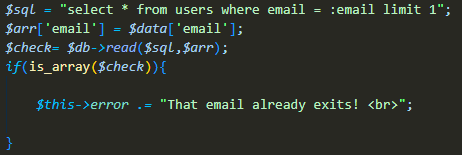
E tambem validação da palavra pass e verificação se a primeira palavrapass é igual ou não a segunda palavra pass:



Verificação se a palavra pass é mais de 4 caracteres: 

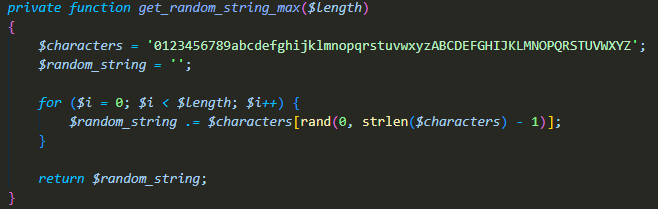
caso não seja maior que 4 caracteres o programa escreve uma mensagem de erro a informar que a palavra pass tem que ser mais de 4 caracteres.

Verifica se o email já esta a ser utilizado :



Caso esteja já em uso escreve uma mensagem de erro a informar que o email fornecido já esta em uso.

Cria um numero aleatório para quando um utilizador fazer um novo registro seja atribuído um numero como se fosse um identificação dentro da base de dados



Para facilitar a implementação da aplicação recorri alguma documentação online, visto que, este tipo de matérias não são lecionadas durante o nosso curso.

Optei por escolher este método de desenvolvimento pois assim consigo ter uma precessão melhor e estruturado do meu código, atribuindo responsabilidades a cada classe e a cada método.

A Aplicação é dividida em duas partes “app”,”public”, cada parte tem a sua função ,a zona “public”

É dividido e quatro partes começando a inicialização do programa com o ficheiro “index.php” que com o comando “session\_start();”



Inicia uma sessão PHP ou retoma a sessão atual baseada em um identificador de sessão passado via um pedido GET, POST ou um cookie. As sessões permitem armazenar informações entre diferentes páginas e pedidos.

**Define a constante ‘ROOT’**

o código constrói o URL base do projeto e define a constante ‘ROOT’ com ele.

* ‘$\_SERVER['REQUEST\_SCHEME']’: Protocolo utilizado (‘http’ ou ‘https’).
* ‘$\_SERVER['SERVER\_NAME']’: Nome do servidor (domínio).
* ‘$\_SERVER['PHP\_SELF']’: Caminho do script atual.

Depois, o código remove "index.php" do caminho para obter o diretório base e define a constante ROOT com este valor.

**Define a constante ‘ASSETS’**

constante ‘ASSETS’ como a URL do diretório de ativos do projeto, concatenando a URL base (‘$path’) com "assets/".

**o arquivo init.php**

Inclui o arquivo ‘init.php’ localizado no diretório’ ../app/’. Esse arquivo provavelmente contém definições e configurações iniciais necessárias para o funcionamento do aplicativo.

**Inicia a aplicação**

A última linha instancia a classe App e cria um novo objeto $app. Presumivelmente, a classe App está definida no arquivo init.php ou em algum arquivo incluído por ele. Esta classe provavelmente controla a lógica principal da aplicação, como roteamento de URLs, controle de fluxo da aplicação, etc.

**‘ASSETS’**

O Ficheiro ‘ASSETS’ em um programa armazena todos os recursos estáticos necessários, como imagens, áudio, vídeos, fontes, estilos (CSS), e dados (JSON, XML). Ela organiza e centraliza esses arquivos para facilitar o gerenciamento e o uso no desenvolvimento do software.

**‘UPLOADS’**

**Função:** Armazenamento de arquivos enviados por usuários.

**Utilização:** Temporário ou permanente para processamento e compartilhamento.

**Considerações:** Segurança e gerenciamento de acesso são essenciais.

( A pasta upload só é criada após o carregamento da primeira imagem caso não exista)

**A ZONA ‘APP’**

A zona ‘app’ é constituída por 4 partes : ‘core’,’controllers’,’views’,’models’

**CORE:**

* App.php: app.php define a classe App que controla o roteamento dinâmico de URLs em um aplicativo web, carregando e executando métodos de controladores com base nas URLs solicitadas pelos usuários.
* Controller.php: O arquivo controller.php define a classe Controller, que facilita o carregamento de views (view) e modelos (load\_model). Ele garante que as views sejam carregadas corretamente e trata casos onde a view não existe, mostrando uma página de erro (404.php).
* Database.php: database.php define a classe Database com conexão PDO, seguindo o padrão Singleton para garantir uma única instância. Oferece métodos (read() e write()) para consultas SQL e tratamento de exceções para erros no banco de dados
* Function.php:

**show($data)**: Imprime dados formatados para depuração usando print\_r() dentro de tags <pre>.

**check\_error()**: Verifica e exibe mensagens de erro armazenadas na variável de sessão $\_SESSION['error'], removendo-a em seguida.

**CONTROLLER.PHP:**

* **Login.php:**O programa PHP no arquivo login.php controla a exibição da página de login e processa o login do usuário quando os dados são enviados através do método POST. Ele utiliza um modelo de usuário (User) para executar o processo de login.
* **Logout.php:** O arquivo logout.php define um controlador PHP chamado logout que gerencia o processo de logout do usuário. Ele carrega o modelo User e utiliza o método logout() desse modelo para realizar o logout do usuário.
* **Register.php:** O ficheiro register.php serve para permitir o registo de novos utilizadores num site. Ele apresenta um formulário onde os utilizadores podem introduzir informações como nome, email e senha para criar uma conta. Este ficheiro é fundamental para a funcionalidade de registo e para a integração de novos membros na plataforma online.

faz o arranque do programa

Para o funcionamento do site e das ações nele pedidas pelo utilizador o programa inicia com o “índex.php” que inicia a sessão,constrói a URL base do projeto, define as constantes para o caminho raiz e para os ativos (assets) da aplicação, inclui um arquivo de inicialização (init.php) e, de seguida, instancia a classe App para iniciar a aplicação.que é o arranque do site seguida do “app.php”  ele é responsável por analisar a URL fornecida, determinar o controlador e o método a serem chamados com base na URL e seus parâmetros, e, de seguida, chamar o método apropriado do controlador e carrega a view desejada pelo utlilizador.

Ao elaborar o relatório devem ter em atenção que o relatório funcionará como documentação do projeto desenvolvido. Dessa forma devem-se perguntar se o relatório descreve de forma suficientemente detalhada o trabalho para que possa ser compreendido e reproduzido se necessário no futuro por outras pessoas.

*O aluno deve, conjuntamente com o professor orientador, definir a estrutura mais adequada ao seu projeto.*

# Análise do Sistema

Análise das especificações do problema; identificação das entidades e das suas relações, dos fluxos de dados, esquema de tabelas da base de dados, dicionário de dados, definição de interfaces e layout, etc. Desenvolvimento do modelo conceptual para resolução do problema.

# Desenvolvimento

Este capítulo descreve a implementação da solução proposta no capítulo anterior.

Neste capítulo são também descritas as especificidades de implementação de acordo com o ambiente de desenvolvimento, plataforma e linguagem escolhida para o desenvolvimento. Devem também ser reportados os problemas encontrados e a solução escolhida para os resolver.

Descrição concisa do hardware e do software utilizado.

# Avaliação

Breve descrição de como instalar, usar ou aceder ao sistema desenvolvido. Se a descrição for extensa, considerar a sua localização num anexo ou mesmo num documento à parte. Descrição dos testes realizados e os resultados experimentais obtidos; análise crítica dos resultados.

Este capítulo pode não existir em todos os projetos, dependendo da sua natureza. Em alguns casos é também aceitável que se transforme numa secção do capítulo anterior.

# Conclusão

O capítulo de conclusões é um dos mais importantes do relatório, sendo aqui que devem ser apresentados os resultados do trabalho elaborado.

Deve conter uma síntese do trabalho, com os resultados mais importantes e a sua relação com os objetivos propostos e com os meios usados.

A conclusão geral do trabalho deve apresentar recomendações e sugestões, resultantes do trabalho realizado, sempre que tal se aplicar.

As conclusões finais devem focar o sucesso/insucesso do trabalho, revendo as dificuldades encontradas. Devem resumir, de alguma forma, as vantagens do produto desenvolvido. Podem também referir as facilidades e as dificuldades sentidas ao longo do projeto.

Deve também apresentar sugestões para a continuação do trabalho.

A conclusão deve dar resposta as seguintes questões:

Em que medida os objectivos foram alcançados?

Quais foram as lições aprendidas?

Quais são as ideias para trabalhos futuros?

Quais são as vantagens de desvantagens da solução apresentada, face a outras já existentes?

# Bibliografia

Esta secção lista as referências bibliográficas citadas no texto.

Deverá haver citação de uma referência sempre que se utilizam ideias, conhecimentos ou métodos que não são da nossa autoria.

Os itens da lista de referências poderão ser listadas por ordem alfabética ou de ocorrência no texto e a ordem é identificada por um número entre parênteses, de acordo com as seguintes normas:

1. As referências bibliográficas **Livros (monografias):**

*Um autor:*

1. APELIDO, nome - Título. N.º edição. Local de publicação: Editor, Ano.

FRADA, J.J.C - Guia Prático para a elaboração de Trabalhos Científicos. 2ª ed. Lisboa: Editorial Cosmos, 1993.

*Dois autores até três:*

1. BOURDIEU, P.; PASSERON, J.C. - A Reprodução – elementos para uma teoria do

sistema de ensino. 1ª ed. Lisboa: Editorial Veja, 1970.

*Mais do que três autores:*

1. CHEVALIER, Brigitte [et al] – Semântica e Pragmática do Discurso. 3ª ed. Coimbra: Almedina, 1998.
2. **Fontes informáticas (base de dados, CD-ROM, Sites, etc.)**

Título: Complemento do título. Responsabilidade, Ano. [fonte informática].

Hotelaria em Portugal: Evolução das qualificações e diagnóstico das necessidades de formação. INOFOR,1999. [CD-ROM]

Teste em grupo a impressoras de jacto de tinta. PC Guia. Dir. Cristina Magalhães. XL, 2003. [Site] Acesso: <http://pcguia.xl.pt/pcg/1003/hardware/a01-00-00.shtml>.

(No caso de sites deve acrescentar-se à referência a palavra Acesso: [seguida do endereço da página.]).

Nota: Se o documento não possuir o local, nome da editora e/ou data da publicação, deverão utilizar-se as abreviaturas: s.l. (sem local); s.n. (sem nome da editora); s.d. (sem data).

1. **Publicações em série (revistas, jornais…)**

Titulo: Complemento do Titulo. Responsabilidade. Edição. Numeração.

Local de publicação: Editor, Ano.

Mini-Micro’s: revista de computadores, vídeo e som. Dir. Luís Oliveira.

Ano, n.º 1(maio 1983). Lisboa: Socedite, 1983.

Anexos

Algo que completa o texto mas que, se fosse inserido no corpo do relatório, romperia a apresentação lógica e ordenada do trabalho.

Algo que não pode ser colocado com facilidade no corpo do relatório devido à sua dimensão ou ao tipo de reprodução utilizado.

Algo que pode ser omisso para uma leitura menos técnica, mas que é necessário para uma análise mais aprofundada.

Os eventuais elementos que podem figurar num anexo são:

* Desenhos, esquemas e quadros suplementares;
* Fotografias;
* Ilustrações;
* Listas de programas de computador;
* Mapas;
* …

Cada anexo deverá começar numa nova página e a sua identificação será feita por intermédio de uma letra maiúscula na sequência do alfabeto (Anexo A, Anexo B,...) seguida do respetivo título:

# Anexo A - Listagem do material utilizado

# Anexo B - Software