

# Atualização de Aplicação Web de Gestão de Processos de Formação

João Carlos Peixoto Freitas  
(nrº 16968, regime diurno)

Orientação de  
Joaquim G. P. Silva

LICENCIATURA EM ENGENHARIA EM SISTEMAS INFORMÁTICOS  
ESCOLA SUPERIOR DE TECNOLOGIA  
INSTITUTO POLITÉCNICO DO CÁVADO E DO AVE

## **Identificação do aluno**

João Carlos Peixoto Freitas  
Aluno número 16968, regime diurno  
Licenciatura em Engenharia em Sistemas Informáticos

## **Orientação**

Joaquim G. P. Silva

## **Informação sobre o Estágio**

NKA - New Knowledge Advice, Lda  
Travessa José Cruz, nº5, 4715-343 Braga, Portugal  
Engenheiro João Magalhães

## Resumo

Não é segredo para ninguém que, a fim de manter o negócio competitivo no mercado, é preciso apostar em inovação. Por meio dessa iniciativa, a empresa consegue criar soluções eficientes, minimizar o impacto de eventualidades, antever situações adversas e gerar alternativas que proporcionem experiências positivas ao consumidor.

Uma empresa que acredita que o seu produto já é bom o suficiente ao ponto de não precisar de inovação está destinada ao fracasso. Adiar ao máximo a atualização tecnológica, ou somente adotar medidas quando o mercado exige pode resultar numa perda de competitividade.

Tendo isto em consideração, com este trabalho foram atualizadas as tecnologias de um produto de forma a manter a sua operabilidade na tecnologia mais atual possível. Integraram-se no NkaAcademies, um produto desenvolvido pela NKA - *New Knowledge Advice*, atualizações ao nível de linguagem de programação e bibliotecas utilizadas de forma a tornar o produto o mais atual possível em termos tecnológicos e foram ainda desenvolvidas novas funcionalidades para adicionar futuramente ao produto.



## Abstract

It is no secret that, in order to keep the business competitive in the market, it is necessary to invest in innovation. With that in mind, the company is able to create efficient solutions, minimize the impact of eventualities, anticipate adverse situations and create alternatives that provide positive experiences for the costumer.

A company that believes its product is good enough, that it doesn't need innovation is doomed to failure. Postponing the technological update as much as possible, or only adopting measures when the market demands it, can result in a loss of competitiveness.

Keeping this in mind, with this work the technologies of a product were updated in order to maintain its operability in the most current technology possible. In addition to NkaAcademies, a product developed by NKA - *New Knowledge Advice*, updates to the programming language and libraries used were made in order to make the product as up-to-date as possible in terms of technology, and the development of new functionalities to be added to the product in the future.



## Agradecimentos

O desempenho no desenvolvimento deste projecto de estágio não teria sido o mesmo sem a ajuda e apoio de algumas pessoas, às quais gostaria de aqui expressar o meu reconhecimento.

Assim, gostaria de começar por agradecer à empresa NKA - New Knowledge Advice pela oportunidade de realizar estágio numa época complicada que vivemos e por toda a ajuda ao longo do estágio. Um obrigado em especial ao meu orientador na empresa, Eng. João Magalhães, pelo acompanhamento ao longo do meu trabalho e pela disposição de transmitir valores profissionais. Aproveito também para agradecer à restante equipa da NKA pela simpatia e familiaridade durante este percurso.

Gostaria de agradecer também ao meu orientador de estágio, o professor Joaquim Silva pela orientação, disponibilidade e conselhos dados ao longo desta etapa.

Por fim, agradeço à minha família e amigos por todo o apoio pois é também graças a eles que consegui acabar esta etapa da minha vida.





# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Objetivos . . . . .	1
1.2	Contexto . . . . .	1
1.3	Plano de trabalhos . . . . .	1
1.4	Estrutura do documento . . . . .	2
<b>2</b>	<b>Processo de Reengenharia</b>	<b>5</b>
2.1	Sistemas AS-IS . . . . .	5
2.1.1	Descrição da Arquitetura/tecnologias utilizadas . . . . .	5
2.2	Sistemas TO-BE . . . . .	6
2.2.1	Identificação das lacunas/melhorias a realizar . . . . .	6
2.2.2	Requisitos . . . . .	6
2.2.3	Alterações no desenvolvimento da solução . . . . .	6
<b>3</b>	<b>Desenvolvimento da aplicação</b>	<b>11</b>
3.1	Análise e Especificações . . . . .	11
3.1.1	Requisitos Funcionais . . . . .	11
3.1.2	Requisitos Não Funcionais . . . . .	12
3.2	Arquitetura técnica . . . . .	12
3.2.1	Diagrama de Atividades . . . . .	13
3.2.2	Estrutura do código . . . . .	14
<b>4</b>	<b>Análise dos Resultados</b>	<b>17</b>
4.1	Sub-capítulo 1 . . . . .	17
4.2	Sub-capítulo 2 . . . . .	17
<b>5</b>	<b>Conclusão Final</b>	<b>19</b>
5.1	Sub-capítulo 1 . . . . .	19
5.2	Sub-capítulo 2 . . . . .	19



# Lista de Figuras

1.1	Diagrama de Gantt . . . . .	2
3.1	Diagrama de Atividades . . . . .	13
3.2	Estrutura do código . . . . .	14
3.3	Interface Web - template teste . . . . .	16



# Lista de Tabelas

3.1	Requisitos funcionais . . . . .	12
3.2	Requisitos não funcionais . . . . .	12



# Listagens de Código

2.1	Função <code>mysql_select_db</code> . . . . .	7
2.2	Função <code>mysqli_select_db</code> . . . . .	7
2.3	Função <code>mysql_result</code> . . . . .	7
2.4	Função para substituir <code>mysql_result</code> . . . . .	8
2.5	Função <code>mysql_field_name</code> . . . . .	8
2.6	Função para substituir <code>mysql_field_name</code> . . . . .	8
2.7	Improvement da função <code>nka_mysql_result</code> . . . . .	9
3.1	Exemplo de request à base de dados . . . . .	15
3.2	Função JavaScript utilizando <code>jQuery.ajax()</code> . . . . .	15





# Siglas & Acrónimos

**HTML** HyperText Markup Language. 15

**PHP** HyperText Preprocessor. 5, 6, 9, 14

**SQL** Structured Query Language. 7–9, 14, 15

**WYSIWYG** *What You See Is What You Get*. 16

**XAMPP** Cross-Plataform(X), Apache(A), MariaDB(M), PHP (P) e Perl (P). 5



# Glossário

**Bootstrap** *Framework* web para desenvolvimento de componentes de interface e front-end para sites e aplicações web usando HTML, CSS e JavaScript. 12, 15

**MoSCoW** Método de priorização de funcionalidades ou requisitos. 11



# 1. Introdução

Acompanhar as novas tecnologias é fundamental para um negócio. É preciso investir em inovação para manter um negócio no mercado competitivo. A tecnologia, aliada a uma boa atualização, consegue proporcionar soluções eficientes às empresas.

– por terminar – acabaram as ideias !!

## 1.1 Objetivos

Este estágio está diretamente relacionado com o produto NkaAcademies, que se trata de uma solução web completa para a gestão de processos de formação.

Pretende-se efetuar a migração completa do produto NkaAcademies para as versões mais recentes do PHP e do MySQLi, no entanto será necessário resolver alguns problemas de compatibilidade que poderão surgir consequentemente a esta alteração e numa fase mais avançada do estágio, incorporar novas funcionalidades ao produto.

## 1.2 Contexto

Este projeto foi realizado no âmbito de estágio curricular da Licenciatura em Engenharia de Sistemas Informáticos, na empresa NKA - *New Knowledge Advice* em Braga.

O estágio curricular teve uma duração de aproximadamente quatro meses, tendo iniciado a 22 de fevereiro e terminado a 9 junho de 2021, de segunda a quinta feira num regime maioritariamente de teletrabalho, com exceção de algumas semanas em regime presencial.

A NKA - *New Knowledge Advice* é uma empresa tecnológica sediada em Braga, fundada em 2011 que concebe e desenvolve soluções globais vocacionadas para a otimização de cada negócio, por meio de projetos específicos, fornecendo as soluções mais adequadas às crescentes exigências dos mercados. A empresa dedica-se ainda ao desenvolvimento de software, à implementação de sistemas e à consultoria e formação(NKA, 2021).

## 1.3 Plano de trabalhos

No âmbito do estágio, foram identificadas as seguintes tarefas:

- Conhecer a empresa, tecnologias e projeto a desenvolver - Migração do produto NkaAcademies para as versões mais recentes do PHP e MySQLi;

- Investigação sobre PHP e MySQLi - Leitura de CHANGELOG das versões anteriores. Análise de funções *deprecated*.
- Alteração do código para a versão 8 PHP.
- Testes e correção de erros de compatibilidade
- Conhecer o projeto a desenvolver - módulo auxiliar ao processo de gestão de formação.
- Levantamento de requisitos e funcionalidades
- Desenvolvimento do módulo - Filtro de pesquisa e alteração manual e automática de documentos associados à gestão de formação.
- Testes e correção de erros
- Preparação da aplicação - Alteração de ligações para cloud. Limpeza de código.
- Atualização das alterações no servidor

Para planear a execução das tarefas, foi elaborado um diagrama de Gantt de acordo com a Figura - 1.1.

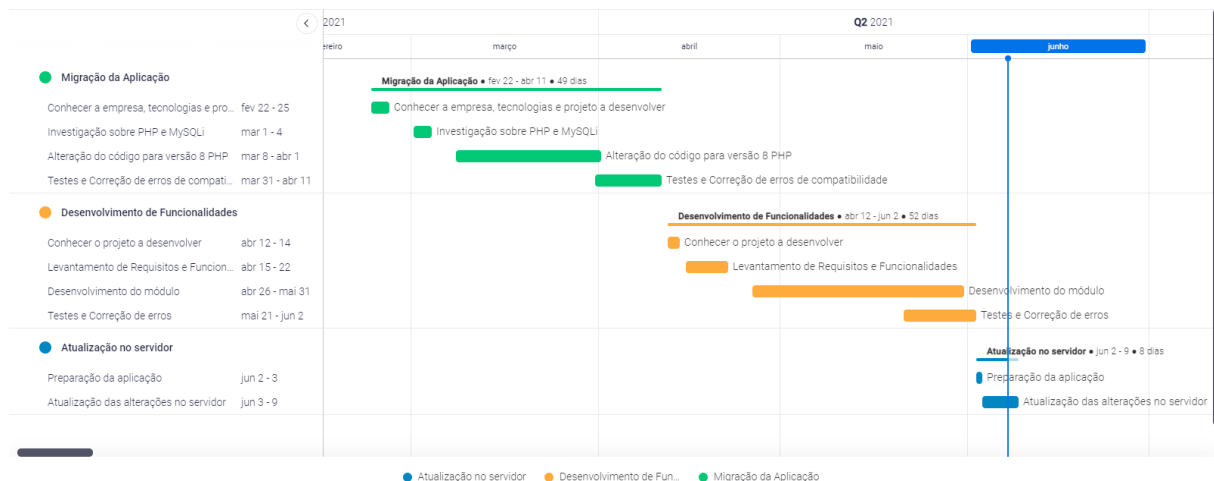


Figura 1.1: Diagrama de Gantt

## 1.4 Estrutura do documento

[A última secção da introdução deve explicar a estrutura do documento: quais são só capítulos existentes (para além do primeiro) e o que será discutido em cada um desses capítulos. A estrutura típica de um relatório de desenvolvimento de software é:

Introdução, com um breve resumo do que se pretende atingir, e uma descrição clara dos objetivos;

1. Análise ao problema, que poderá incluir uma análise ao estado da arte ou ao modelo de negócio onde se pretende intervir;

2. Análise e modelação do sistema, em que sejam levantados sistematicamente os requisitos, descritos diagramas de caso de uso e de atividade (que descrevam/formalizem o modelo de negócio).
3. Implementação, em que se descrevam as tecnologias escolhidas (e se justifiquem), e se refira detalhes sobre a implementação.
4. Análise de resultados e testes, seja uma análise/avaliação aos resultados obtidos, sejam testes de usabilidade ou unitários ao trabalho desenvolvido.
5. Conclusão.]





## 2. Processo de Reengenharia

### 2.1 Sistemas AS-IS

#### 2.1.1 Descrição da Arquitetura/tecnologias utilizadas

##### Tecnologias e Plataformas

Neste ponto serão descritas as tecnologias e plataformas utilizadas ao longo deste trabalho, tais como XAMPP, Laragon, Visual Studio Code.

Para a realização deste projeto foram utilizadas as plataformas e tecnologias requeridas pela empresa para facilitar a implementação e integração do projeto, e para tal foi necessário proceder à sua aprendizagem.

De seguida vão ser descritas de forma sucinta as tecnologias e plataformas utilizadas ao longo do projeto.

##### XAMPP

O XAMPP é formado por um pacote que inclui, base de dados MySQL, servidor web Apache e interpretadores para as linguagens de script. É essencialmente utilizado pelos desenvolvedores que pretendem criar um servidor web local no seu próprio computador, com a finalidade de realizar testes sem necessitar de acesso à rede (Friends, 2018).

##### Visual Studio Code

O VsCode é um editor de código-fonte simplificado com suporte para operações de desenvolvimento como *debugging*, execução de tarefas e controlo de versões (Microsoft, 2021).

##### Xdebug

O Xdebug é uma extensão para PHP que fornece uma variedade de recursos para melhorar a experiência de desenvolvimento de PHP (Rethans, 2020).

##### Laragon

O Laragon é uma maneira rápida e fácil de criar um ambiente de desenvolvimento isolado no Windows. Inclui Mysql, PHP Memcached, Redis, Apache (Keycdn, 2015).

## 2.2 Sistemas TO-BE

### 2.2.1 Identificação das lacunas/melhorias a realizar

Atualmente, no mundo em que vivemos, *updates* tecnológicos são lançados praticamente todos os dias relativos a sistemas de informação. É do maior interesse da parte de uma empresa, manter o seu produto atual e competitivo.

De momento, o produto NkaAcademies encontra-se desatualizado, utilizando versões antigas de PHP(5.6) e MySQL. Foi então proposto desenvolver a migração de todo o produto NkaAcademies para as versões mais recentes, suportadas no *host* utilizado pela empresa, do PHP(5.6 -> 8.0.2) e do MySQL com o objetivo de atualizar o produto e também, em casos futuros o uso de novas funcionalidades que surgiram apenas em versões superiores à que o produto estava desenvolvido.

As melhorias a realizar passaram maioritariamente pela substituição total da biblioteca PHP MySQL para MySQLi, uma vez que grande parte das funções que fazem parte da biblioteca MySQL foram descontinuadas / *deprecated*, enquanto noutros casos apenas foram feitas alterações ao nível da chamada da função ou dos parâmetros (e ordem) recebidos. Posto isto, procedeu-se à sua alteração total no produto NkaAcademies, sendo que no caso de funções *deprecated* foram implementadas alternativas *hard-code* em PHP.

### 2.2.2 Requisitos

Uma vez que se trata de uma migração completa de um produto pré-existente e que apenas foram efetuadas alterações ao nível de chamadas de funções, parâmetros recebidos por funções e correções de erros de compatibilidade conclui-se que os requisitos são os mesmos que foram inicialmente estipulados para o desenvolvimento do produto NkaAcademies.

Como tal, o requisito mais importante definido para o cumprimento desta fase tem a ver com a compatibilidade das alterações feitas. As novas funções e todas as modificações realizadas ao nível do produto devem ser implementadas tendo em vista a sua compatibilidade com as versões anteriores, que neste caso são as versões PHP 5.6 e 7 e a compatibilidade com outras funcionalidades já existentes no produto garantindo a operabilidade e o bom funcionamento da aplicação na nova versão tecnológica.

### 2.2.3 Alterações no desenvolvimento da solução

Neste capítulo serão apresentadas algumas das principais alterações implementadas no produto NkaAcademies na fase da migração da aplicação para as versões mais recentes do PHP e MySQL.

As principais alterações efetuadas no projecto NkaAcademies foram ao nível da chamadas de funções, implementação de novos métodos devido à descontinuidade de funções da biblioteca MySQL, parâmetros (e ordem) recebidos por funções e correções de erros de compatibilidade entre versões.

*mysql\_select\_db vs mysqli\_select\_db*

Neste caso, será apresentado um excerto de código relativo a uma função que foi atualizada na biblioteca MySQLi, recebendo o mesmo tipo de parâmetros mas que trocou a ordem de como eram incorporados na função.

A função *mysql\_select\_db* (listagem 2.1) seleciona uma base de dados do tipo MySQL.

```
function mysql_select_db($database , $link_identifier=NULL):bool
```

Listagem 2.1: Função *mysql\_select\_db*.

A função da Listagem 2.1 recebe como argumentos:

- **\$database**: nome da base de dados que deverá ser selecionada.
- **\$link\_identifier**: conexão ao MySQL.

A função *mysqli\_select\_db* (listagem 2.2) seleciona a base de dados a ser utilizada ao realizar *queries* do tipo MySQLi.

```
function mysqli_select_db($mysql , $database):bool
```

Listagem 2.2: Função *mysqli\_select\_db*.

A função da Listagem 2.2 recebe como parâmetros:

- **\$mysql**: *connection string* de acesso à base de dados.
- **\$database**: nome da base de dados que deverá ser selecionada.

*mysql\_result vs nka\_mysqli\_result*

Neste exemplo será apresentado nos excertos de código a alteração que teve de ser feita quando a função da biblioteca MySQL não foi continuada na biblioteca MySQLi, fazendo com que tivesse de ser criada uma equivalente.

A função *mysql\_result* (listagem 2.3) obtém os dados de um determinado resultado.

```
function mysql_result($result , $row , $field=0):string
```

Listagem 2.3: Função *mysql\_result*.

A função da Listagem 2.3 recebe como argumentos:

- **\$result**: resultado da *query* SQL a ser executado.
- **\$row**: número da linha do resultado.
- **\$field**: nome do campo a ser procurado (tabela).

A função *nka\_mysqli\_result* (listagem 2.4), utilizada para substituir a função *deprecated mysql\_result* foi implementada recebendo os mesmos argumentos que a anterior, e retornando uma *string*.

```

    if (!function_exists('nka_mysql_result')) {
        function nka_mysql_result($res, $row, $field=0) {
            $res->data_seek($row);
            $datarow = $res->fetch_array();
            return $datarow[$field];
        }
    }

```

Listagem 2.4: Função para substituir mysql\_result.

A função da Listagem 2.4 recebe como argumentos:

- **\$result**: resultado da *query* SQL a ser executado.
- **\$row**: número da linha do resultado.
- **\$field**: nome do campo a ser procurado (tabela).

### *mysql\_field\_name vs mysqli\_field\_name*

Nas Listagens 2.5 e 2.6 está mais uma vez demonstrado o caso em que uma função da biblioteca MySQL é descontinuada para MySQLi e teve de se proceder à sua implementação.

A função *mysql\_field\_name* (listagem 2.5) obtém o nome do campo especificado num resultado.

```

function mysql_field_name($result, $field_offset):string|false

```

Listagem 2.5: Função mysql\_field\_name.

A função da Listagem 2.5 recebe como parâmetros:

- **\$result**: resultado da *query* SQL a ser executada.
- **\$field\_offset**: index/*offset* de um campo numérico.

A função *mysqli\_field\_name* (listagem 2.6), utilizada para substituir a função *deprecated mysql\_field\_name* foi implementada no projeto recebendo os mesmos parâmetros que a anterior, e retornando uma *string* ou *false*.

```

    if (!function_exists('mysqli_field_name')) {
        function mysqli_field_name($result, $field_offset){
            $properties = mysqli_fetch_field_direct($result, $field_offset);
            return is_object($properties) ? $properties->name : false;
        }
    }

```

Listagem 2.6: Função para substituir mysql\_field\_name.

A função da Listagem 2.6 recebe como parâmetros:

- **\$result**: resultado da *query* SQL a ser executada.
- **\$field\_offset**: index/*offset* de um campo numérico.

**Improvement da função *nka\_mysql\_result***

Neste caso está demonstrado um exemplo *improvement* de uma função que já tinha sido desenvolvida para substituir uma função descontinuada da biblioteca MySQL.

A função *nka\_mysql\_result\_v2* (listagem 2.7), foi um *improvement* da função previamente implementada *nka\_mysql\_result*. Esta alteração deveu-se ao facto de alguns erros na incompatibilidade das diferentes versões(7 e 8) do PHP em que o *\$datarow* chegava *empty*, enquanto que na versão PHP5.6 a mesma função não apresentava erro. Verificou-se que a mesma função se comportava de forma diferente de versão para versão e como tal, procedeu-se à sua remodelação passando agora a retornar *string* ou *null* resolvendo o problema.

```

    if (!function_exists('nka_mysql_result_v2')) {
        function nka_mysql_result_v2($res, $row, $field=0) {
            $res->data_seek($row);
            $datarow = $res->fetch_array();
            $retval = null;
            $cond = ! empty($datarow);
            if($cond){
                $retval = $datarow[$field];
            }
            return $retval;
        }
    }

```

Listagem 2.7: Improvement da função *nka\_mysql\_result*.

A função da Listagem 2.7 recebe como argumentos:

- **\$result**: resultado da *query* SQL a ser executado.
- **\$row**: número da linha do resultado.
- **\$field**: nome do campo a ser procurado (tabela).



## 3. Desenvolvimento da aplicação

Este capítulo apresenta o trabalho desenvolvido numa segunda parte do estágio, o desenvolvimento de um módulo a implementar no produto NkaAcademies. Foi necessário analisar requisitos, tratar da modelação do produto e da sua implementação.

### 3.1 Análise e Especificações

Na identificação de requisitos são enunciadas todas as funcionalidades, de forma enumerada, que se pretendem implementar no *software*. Os diferentes requisitos, funcionais e não funcionais, encontram-se classificados pela metodologia *MoSCoW* que identifica os requisitos de elevada prioridade até aos menos prioritários.

Cada tipo de requisito classificado (*Must*, *Should*, *Could*, e *Would*) representam diferentes significados face ao foco e importância que deverá ser levada em conta durante o processo de desenvolvimento.

Assim sendo:

1. ***Must***: classificados como requisitos mais críticos ou indispensáveis para o produto, pretende-se o total foco sobre este tipo de requisitos durante o processo de desenvolvimento;
2. ***Should***: são tão importantes como os requisitos classificados como *Must Have*, contudo não são considerados tão priorizados;
3. ***Could***: entende-se como requisitos desejáveis, mas também não são necessários;
4. ***Would***: estes são os requisitos menos críticos e com menor valor, podendo ser implementados ou não.

#### 3.1.1 Requisitos Funcionais

Os requisitos funcionais dizem respeito a todas as funcionalidades ou pressupostos de como o sistema deve reagir a entradas específicas e como se deverá comportar. Estes tipos de requisitos podem-se entender como funcionalidades que o utilizador pode interagir de forma direta e mais visível.

A Tabela 3.1 - enuncia os requisitos funcionais estipulados para o sistema.

#	Requisito	Descrição	Prioridade
RF01	Pesquisar formação	O sistema tem que permitir ao utilizador a procura pelo código da formação ou código de curso	MUST
RF02	Escolher declaração	O sistema tem que permitir ao utilizador a escolha da declaração que pretende imprimir / preencher.	MUST
RF03	Editar manualmente	O sistema deverá permitir ao utilizador a edição manual dos campos a preencher	MUST
RF04	Criar campos	O sistema deve permitir a criação de novos campos a serem adicionados às declarações ( <i>template</i> )	COULD
RF05	Editar automaticamente	O sistema deverá permitir ao utilizador a edição automática dos campos a preencher, conforme os resultados da procura	MUST

Tabela 3.1: Requisitos funcionais

### 3.1.2 Requisitos Não Funcionais

Requisitos não funcionais são as propriedades comportamentais relacionadas ao uso da aplicação em termos de desempenho, usabilidade, confiabilidade, segurança, disponibilidade, manutenção e tecnologias envolvidas. Estes requisitos dizem respeito a como as funcionalidades serão entregues ao utilizador do software.

A Tabela 3.2 - encontram-se especificados os requisitos não funcionais do sistema.

#	Descrição	Categoria	Prioridade
RNF01	As novas funções deverão ser implementadas de forma a que sejam compatíveis com outras funcionalidades já existentes	Compatibilidade	MUST
RNF02	O módulo deverá ser implementado em JavaScript, PHP com acesso à base de dados MySQL	Operacional	MUST
RNF03	Apenas utilizadores autenticados e com permissão devem ter acesso módulo desenvolvido	Segurança	MUST

Tabela 3.2: Requisitos não funcionais

## 3.2 Arquitetura técnica

As tecnologias e linguagens utilizadas para o desenvolvimento deste módulo a adicionar ao produto NkaAcademies foram essencialmente descritas no Capítulo 2.1.1, com adição de JavaScript e Bootstrap para desenvolver a componente web da aplicação.



### 3.2.1 Diagrama de Atividades

Para descrever o processo de negócio foi utilizado o diagrama de atividade, uma vez que permite modelar o comportamento do sistema incluindo a sequência e as condições de execução de ações.

A Figura 3.1 demonstra o diagrama de atividades desenvolvido para o sistema.

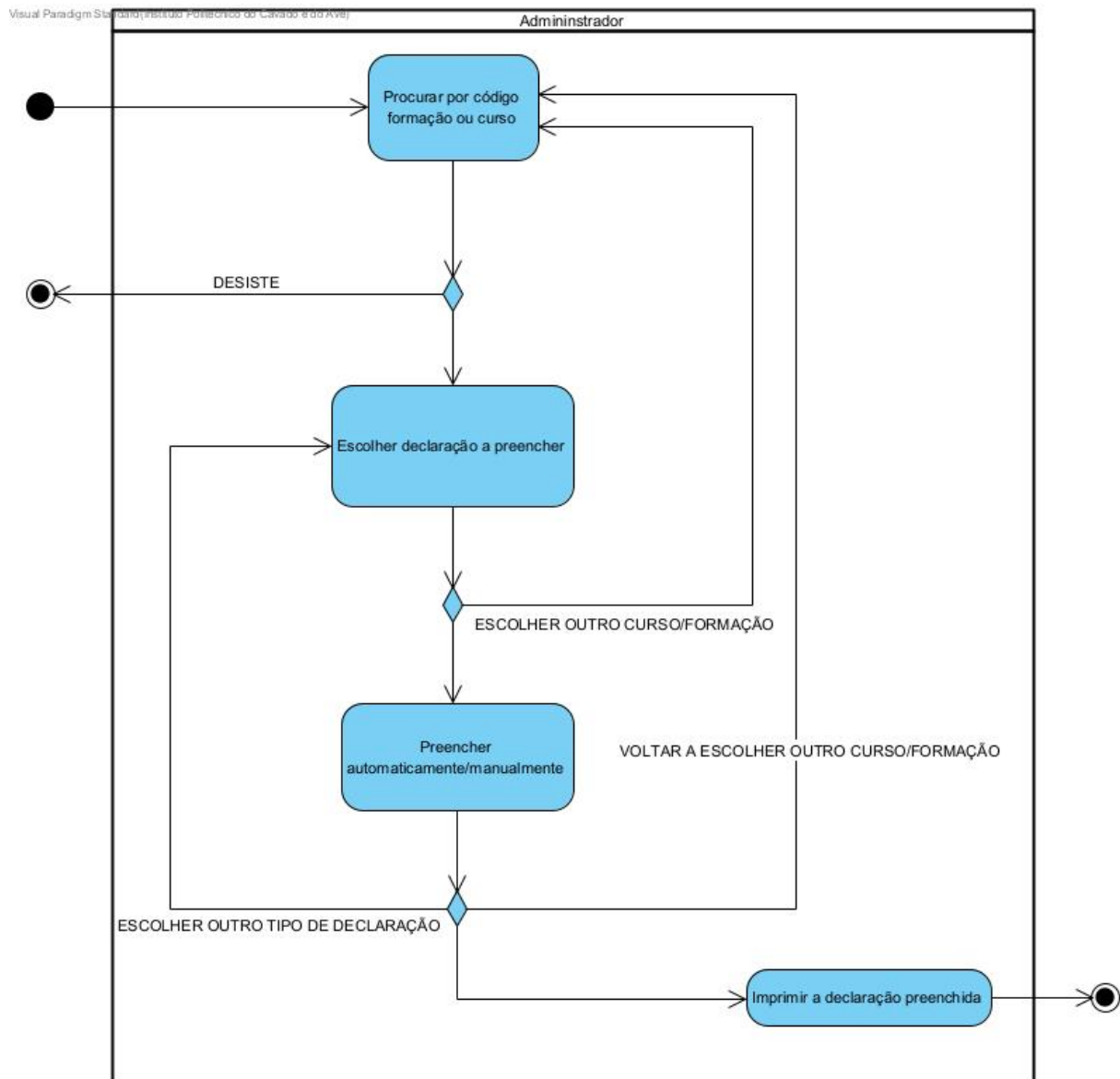


Figura 3.1: Diagrama de Atividades

Recorrendo à visualização do diagrama da Figura 3.1, é possível perceber os diferentes passos que correspondem à atividade geral do desenvolvimento desta fase do projeto, o preenchimento de documentos de apoio aos processos de gestão de formação. As atividades foram então: filtrar a procura a um resultado, escolher um *template* / declaração predefinido, proceder ao seu preenchimento manual/automático e posteriormente à sua impressão caso seja pretendido.

No desenho deste diagrama está subentendido que o cliente está devidamente autenticado e verificadas as suas permissões no acesso a este tipo de ferramenta dentro da

aplicação.

### 3.2.2 Estrutura do código

Neste sub-capítulo será apresentada a estrutura do código desenvolvido, bem como alguns excertos de código para exemplificar as diferentes camadas existentes no projeto: base de dados, *back-end* e *front-end*.

A Figura 3.2 apresenta um esquema gráfico, de forma a que se perceba melhor esta estrutura e as relações entre camadas.

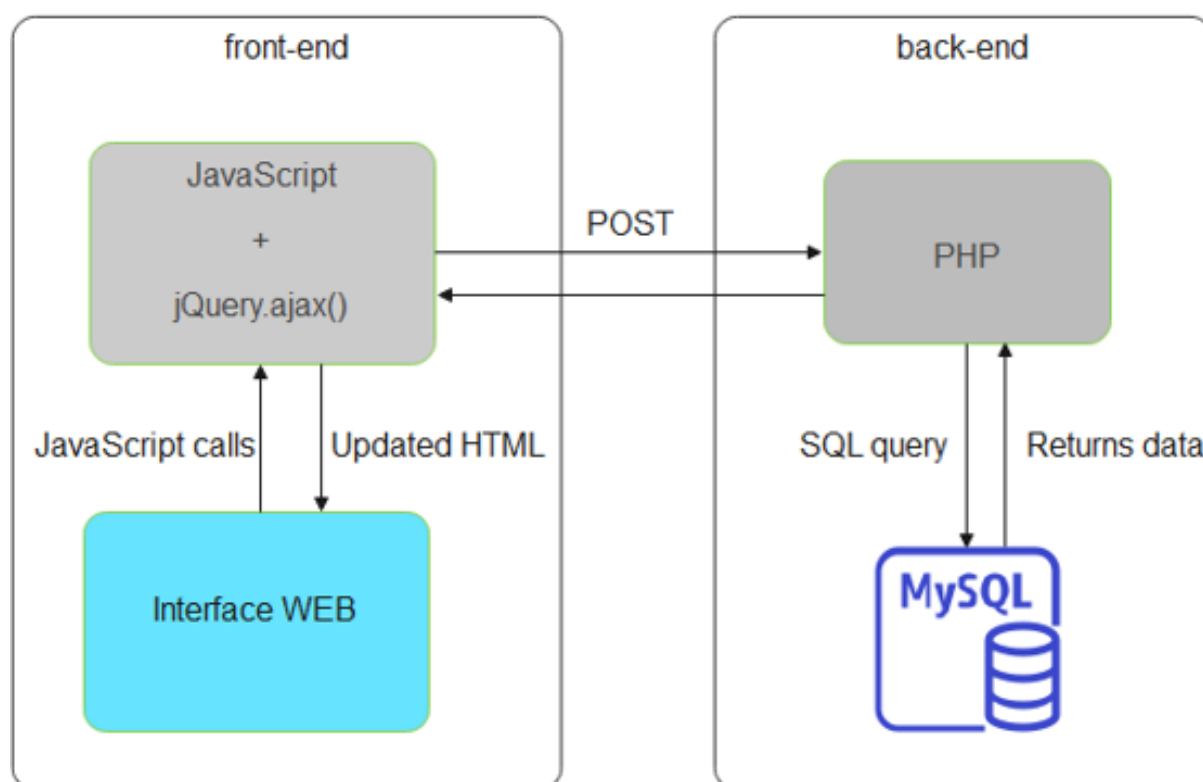


Figura 3.2: Estrutura do código

#### Base de Dados

Quanto à base de dados utilizada para o desenvolvimento do projeto, não foi feito qualquer tipo de modelação uma vez que foi reaproveitada a base de dados já existente do produto NkaAcademies tendo sido apenas necessário acrescentar alguns dados em tabelas(*templates*) já existentes a fim de testes e futura implementação.

#### *back-end*

A parte do *back-end* da aplicação foi desenvolvida em PHP. O principal objetivo do trabalho realizado no *back-end* diz respeito a interações com a base de dados, ou seja, tudo que envolva *queriesSQL* e ligações a bases de dados são definidas neste sector.

Para uma melhor compreensão, será apresentada na Listagem 3.1 um exemplo de uma interação com a base de dados onde basicamente é retornado o conteúdo do *template* selecionado pelo utilizador ( $\$_POST["diploma"]$ ). Caso se verifique que o *iduser* / cliente existe e o template selecionado está definido na base de dados, é devolvido o seu conteúdo para a interface web.

```
case 2073:

$result = mysqli_query($con, "SELECT_*_from_XXX.templates
where_diploma='". $_POST["diploma"] . "' and_iduser='$varemp'");

if (mysqli_num_rows($result)!=0)
{
echo str_replace("table_table-bordered",
"",utf8_encode(nka_mysql_result($result,0,"body")));
}
else
{
        echo "Sem_Template_Definido";
}

mysqli_close($con);

break;
```

Listagem 3.1: Exemplo de request à base de dados

### *front-end*

A parte do *front-end* foi desenvolvida em HTML, recorrendo a *modals* do Bootstrap para implementar a interface web e JavaScript que contém as funções que são o elo de ligação entre a interface web e o *back-end*, que permite atualizar e tornar dinâmica a parte gráfica/visual do sistema.

A Listagem 3.2 demonstra a implementação de uma função JavaScript que complementa a ação realizada ao *back-end* na listagem 3.1.

De salientar ainda, a importância do uso de *jQuery.ajax()* que permite utilizar JavaScript para enviar *asynchronous http request* e obter resposta em vários formatos, e também para atualizar partes de uma página web (que use JavaScript) sem ter que recarregar a página web completa (TutorialsTeacher, 2020).

Assim sendo, o *jquery* neste caso é responsável por passar o parâmetro **diploma** para o *back-end* de forma a ser processada a querySQL correspondente à declaração que o cliente selecionou.

```
function carregatexto(){
    var resultado = $[ 'ajax' ]({
        type: 'POST',
        url: 'phpScripts/dbload.php',
        data: 'kind=2073&diploma='
            +document.getElementById("template").value ,
```

```

    async: false }][ 'responseText ' ];

    $( ".summernote" ).summernote( "code", resultado );

    initcampos ();
}

```

Listagem 3.2: Função JavaScript utilizando jQuery.ajax()

Na Figura 3.3 está representada a interface web resultante das listagens de código descritas acima nos pontos 3.1 e 3.2.

De salientar o uso do *summernote* que é uma biblioteca do JavaScript que ajuda na criação de um editor WYSIWYG online (SummernoteTeam, 2020).

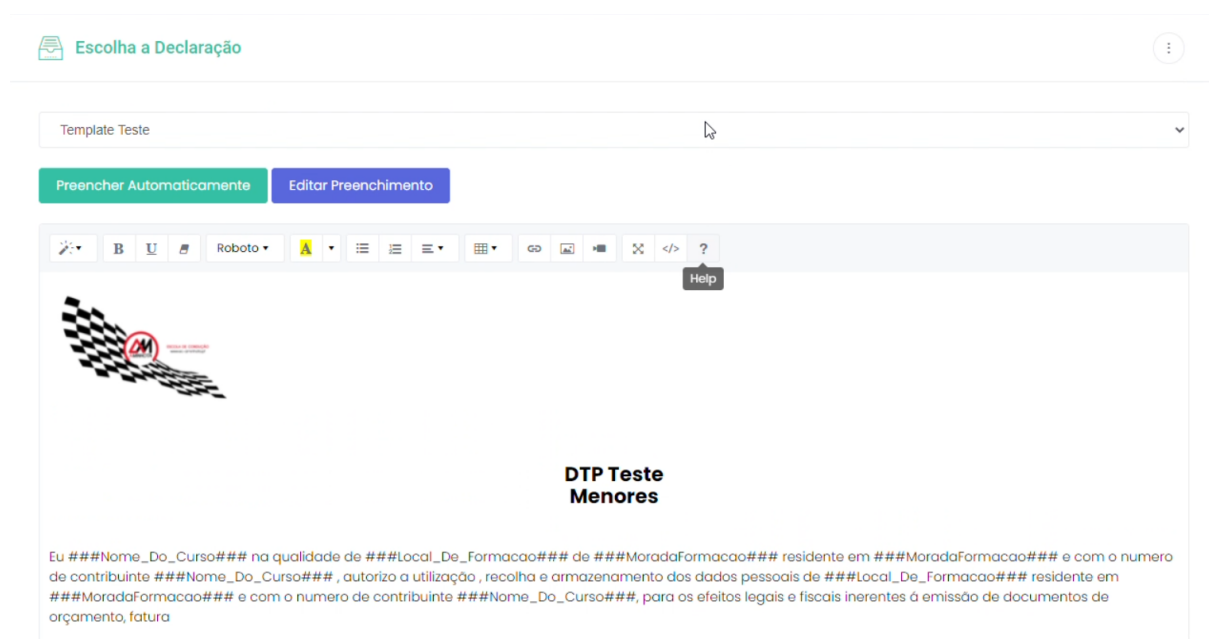


Figura 3.3: Interface Web - template teste

## 4. Análise dos Resultados

### 4.1 Sub-capítulo 1

Texto do sub-capítulo 1

### 4.2 Sub-capítulo 2

Texto do sub-capítulo 2



## 5. Conclusão Final

### 5.1 Sub-capítulo 1

Texto do sub-capítulo 1

### 5.2 Sub-capítulo 2

Texto do sub-capítulo 2





# Bibliografia

- Friends, A. 2018. *About the xampp project*. [https://www.apachefriends.org/pt\\_br/about.html](https://www.apachefriends.org/pt_br/about.html).
- Keycdn. 2015. *100+ awesome web development tools and resources*. <https://www.keycdn.com/blog/web-development-tools>.
- Microsoft. 2021. *Vscode faq*. <https://code.visualstudio.com/docs/supporting/FAQ>.
- NKA. 2021. *Quem somos*. <https://nka.pt/pt-pt/content/quem-somos>.
- Rethans, Derick. 2020. *Xdebug - debugger and profiler tool for php*. <https://xdebug.org/>.
- SummernoteTeam. 2020. *Summernote*. <https://summernote.org/>.
- TutorialsTeacher. 2020. *jquery ajax introduction*. <https://www.tutorialsteacher.com/jquery/jquery-ajax-introduction>.