



Licenciatura em Gestão de Sistemas e Computação

Aplicação web – Comunidade de Robotic Process Automation

Projeto final de Licenciatura

Trabalho realizado por: Bruno Batista Aluno n.º 20182245

Orientador: Professor Doutor Mário Macedo

Barcarena

Novembro de 2021

Universidade Atlântica

Licenciatura em Gestão de Sistemas e Computação

Aplicação web – Comunidade de Robotic Process Automation

Projeto final de Licenciatura

Trabalho realizado por: Bruno Batista Aluno n.º 20182245

Orientador: Professor Doutor Mário Macedo

Barcarena

Novembro de 2021

Resumo

Este trabalho *Website – Comunidade de Robotic Process Automation*, tem como parte fundamental de estudo, a análise dos elementos que envolvem a criação de uma aplicação web.

Tendo perceção diária da realidade que envolve toda a criação da automação de processos robóticos (RPA) e sendo esta uma área em expansão, é necessário haver uma maior quantidade de meios que permitam a troca e partilha de informação de uma forma rápida e colaborativa.

Assim sendo, este estudo tem como objetivo desenvolver uma comunidade virtual que crie interações de pessoas com interesses em comum e contribua para a partilha da informação entre profissionais e aprendizes da área.

Este trabalho baseou-se numa pesquisa bibliográfica sobre o tema apresentado, seguindo-se a descrição, implementação da aplicação que foi desenvolvida e um questionário.

Concluiu-se, com base no questionário online, que as comunidades virtuais são importantes para os profissionais e aprendizes da área que procuram soluções para determinados problemas.

Aferiu-se ainda que este tipo trabalho, tanto a revisão da literatura, como o desenvolvimento e implementação de aplicações *web*, servem nitidamente para adquirir experiências e aprofundar conhecimentos.

Palavras-chave: RPA, automação, empresas, segurança, aplicações *web*.

Abstract

This work Website – Community of Robotic Process Automation, has as a fundamental part of study, the analysis of the elements involving the creation of a web application.

Having daily perception of the reality that involves the whole creation of robotic automation processes (RPA) and this being an expanding area, it is necessary to have a greater amount of means that allow the exchange and sharing of information in a fast and collaborative way.

Therefore, this study aims to develop a virtual community that creates interactions of people with common interests and contributes to the sharing of information between professionals and learners in the area.

This work was based on bibliographic research on the theme presented, followed by the description, implementation of the application that was developed and a questionnaire.

It was concluded that based on the questionnaire, virtual communities are important for professionals and learners in the area who are looking for solutions to certain problems.

It was also pointed out that this type of work, both the literature review, as well as the development and implementation of web applications, serve clearly to acquire experiences and deepen knowledge.

Keywords: RPA, automation, companies, security, web applications.

ÍNDICE

Resumo	4
Abstract.....	5
Índice de figuras	9
Índice de Tabelas.....	11
Abreviaturas e Siglas.....	12
1. Introdução.....	14
1.1. Definição do Problema e Objetivos	15
1.2. Metodologia de Investigação <i>Design Science Research</i>	16
2. Revisão da Literatura.....	17
2.1. RPA.....	17
2.1.1. Vantagens do RPA	17
2.1.2. Desvantagens do RPA	18
2.2. Comunidades virtuais	18
2.3. Engenharia de Requisitos.....	18
2.4. Padrão de arquitetura de software MVC.....	19
2.5. Modelação de base de dados.....	20
2.6. Arquitetura Monolítica	23
2.7. Desenvolvimento WEB	25
2.7.1. HTML.....	25
2.7.2. Web Design	26
2.7.2.1. Web Responsive design	26
2.7.2.2. Adaptive web design	26
2.7.3. PHP.....	26
2.7.3.1. Composer	27
2.8. Segurança com PHP.....	27

2.8.1.	Injeção de SQL	27
2.8.2.	JavaScript	28
2.8.3.	Hash passwords	28
2.8.4.	MYSQL	29
2.8.5.	Bibliotecas	29
2.8.5.1.	AJAX.....	29
2.8.5.2.	jQuery.....	30
2.8.5.3.	Materialize.....	30
2.8.5.4.	Trumbowyg.....	30
2.8.5.5.	API RESTFull Countries.....	31
2.8.5.6.	PHPMailer.....	31
	Análise de requisitos.....	31
3.	Requisitos funcionais da aplicação web	31
3.1.	Requisitos não funcionais	32
3.1.1.	Segurança	32
3.1.2.	Usabilidade	32
3.1.3.	Restrições	32
3.1.4.	Manutenibilidade	33
3.1.5.	Portabilidade e Interoperabilidade.....	33
3.1.6.	Desempenho	33
3.2.	Em seguimento aos requisitos apresentados nas tabelas 1, 2, 3, 4, 5, 6 e 7, foram efetuados os diagramas de caso de uso.....	33
4.	Desenvolvimento da aplicação	35
4.1.	Base de dados.....	36
4.2.	Estrutura das tabelas	36
4.2.1.	Tabela users	36

4.2.2.	Tabela categoria.....	37
4.2.3.	Tabela sub_categoria	37
4.2.4.	Tabela respostas_topico.....	38
4.2.5.	Tabela tópico	39
4.2.6.	Tabela exceções.....	39
4.3.	Desenvolvimento prático da aplicação	40
4.4.	Métodos de acesso ao SGBD.....	40
4.4.1.	Conexão com o SGBD (MySQL).....	40
4.5.	Sistema de rotas	41
4.6.	Tratamento de <i>request</i> via JavaScript.....	43
4.7.	Criação e confirmação de <i>hash</i>	43
4.8.	Exemplo de prevenção de injeção de SQL	45
4.9.	Exemplo para consumir a <i>API REST COUNTRIES</i>	45
4.10.	Apresentação da aplicação web.....	46
4.10.1.	Página inicial.....	46
5.	Demonstração de resultados e análise	52
5.1.	Interpretação dos resultados e conclusões	59
6.	Reflexão final	61
	Bibliografia.....	63
	Anexo A.....	65
	Questionário de opinião sobre a importância das comunidades virtuais e do artefacto desenvolvido.....	65

Índice de figuras

Figura 1 - Exemplo de modelos de dados conceituais (IBM, 2021)	21
Figura 2 - Exemplo de modelos de dados lógicos (IBM, 2021)	22
Figura 3 - Exemplo de modelos de dados lógicos (IBM, 2021)	22
Figura 4 - Arquitetura monolítica de processo único, adaptado de (Newman, 2019)	23
Figura 5 Arquitetura monolítica modular, adaptado de (Newman, 2019)	24
Figura 6 - Exemplo do funcionamento de AJAX (Social Network, sem data)	30
Figura 7 - Diagrama de caso de uso	34
Figura 8 – Estrutura da base de dados	36
Figura 9 - Estrutura do desenvolvimento do projeto em MVC	40
Figura 10 - Ligação com o SGBD e métodos para ligar e desligar da BD	41
Figura 11 - Sistema de rotas	42
Figura 12 - Método para criação de utilizador, com hash da password	43
Figura 13 - Método login	44
Figura 14 – Exemplo de prevenção de injeção de SQL	45
Figura 15 – Exemplo para consumir API REST CUNTRIES via JavaScript	45
Figura 16 - Página inicial da aplicação (index.php)	46
Figura 17 - Modal para criação de um utilizador	47
Figura 18 - Confirmação da conta	47
Figura 19 - Modal para login	48
Figura 20 - Opções do utilizador	48
Figura 21 - Pesquisa de tópicos por datas	49
Figura 22 - Apresenta acesso às categorias e subcategorias	49
Figura 23 - Apresenta tópicos de uma determinada subcategoria	50
Figura 24 - Efetuar posts com preview	50
Figura 25 - Tópicos recentes	51
Figura 26 - Tópicos TOP	51
Figura 27 - Logo da marca, criada em Photoshop	52
Figura 28 - Distribuição dos inquiridos	52
Figura 29 - Idade dos inquiridos	53

Figura 30 - Habilitações dos inquiridos	53
Figura 31 - Resposta à questão número 4	54
Figura 32 - Resposta à questão número 5	54
Figura 33 - Resposta à questão número 6	55
Figura 34- Resposta à questão número 7	55
Figura 35 - Resposta à questão número 8	56
Figura 36 - Resposta à questão número 9	57
Figura 37 - Resposta à questão número 10	57
Figura 38 - Resposta à questão número 11	58
Figura 39 - Resposta à questão número 12	58

Índice de Tabelas

Tabela 1 - Diretrizes - Design-Science Research, adaptado de (Hevner et al., 2004)	16
Tabela 2 - Requisitos funcionais	31
Tabela 3 - Requisitos não funcionais de Segurança	32
Tabela 4 - Requisitos não funcionais de usabilidade	32
Tabela 5 - Requisitos não funcionais de restrições	32
Tabela 6 - Requisitos não funcionais de manutenibilidade	33
Tabela 7 - Requisitos não funcionais de Portabilidade e Interoperabilidade	33
Tabela 8 - Requisitos não funcionais de desempenho	33
Tabela 9 – Campo e atributos da tabela users	36
Tabela 10 – Campo e atributos da tabela categoria	37
Tabela 11 – Campo e atributos da tabela sub_categoria	37
Tabela 12 – Campo e atributos da tabela respostas_topico	38
Tabela 13 – Campo e atributos da tabela tópico	39
Tabela 14 – Campo e atributos da tabela exceções	39

Abreviaturas e Siglas

AJAX - Asynchronous JavaScript and XML

API - Application Programming Interface

CRUD - Create, Read, Update and Delete

CSS - Cascading Style Sheets

HTML - HyperText Markup Language

HTTP - Hypertext Transfer Protocol

HTTPS - Hypertext Transfer Protocol Secure

IT - Information Technology

JS – JavaScript

JSON - JavaScript Object Notation

KPI – Indicadores-Chave de Performance

MVC - Model-View-Controller

PHP - Hypertext Preprocessor

POO - Programação Orientada Por Objetos

PURL - Simple Object Oriented URL

RGPD - Regulamento Geral sobre a Proteção de Dados

ROI – Retorno do Investimento

RPA - Robotic Process Automation

SGBD – Sistema Gestão de Base de Dados

SOAP - Simple Object Access Protocol

SP - Store Procedure

SQL - Structured Query Language

UML - Unified Modelling Language

URL - Uniform Resource Locator

XML - Extensible Markup Language

SGML - Linguagem DeMarcação Generalizada Padrão

DOM - Document Object Mode

1. Introdução

O presente trabalho, foi realizado no âmbito da Monografia da Licenciatura de Gestão de Sistemas e Computação da Universidade Atlântica tendo como tema *Aplicação web – Comunidade de Robotic Process Automation*.

Atualmente as empresas têm a necessidade de entregar valor aos *stakeholders*, para serem mais competitivas e reduzir custos. Mesmo aplicando várias técnicas de gestão nas empresas, também é necessário recorrer às Tecnologias de Informação (TI). Uma das soluções é o recurso à Automação de Processos Robóticos (RPA). O processo de RPA consiste na criação de *robots de software* que permitem simular e executar processos de negócios, baseados nos seus requisitos. Este permite a redução de custos e uma rápida recuperação do Retorno do investimento (ROI), maior rapidez na execução de processos e da sua entrega. Após o processo de automação entrar em produção, este ainda possui uma grande vantagem, pois permite libertar os colaboradores para realizarem outras tarefas essenciais à empresa.

Atendendo ao que foi referenciado, o RPA é uma mais-valia para as empresas, pois permite uma maior eficiência e eficácia das suas operações e melhoria dos Indicadores-Chave de Performance (KPI).

O RPA é uma área das TI em expansão, porém ainda tem pouca partilha de informação *online*, nomeadamente entre profissionais da área.

Ao longo do trabalho foi considerado como mais relevante os seguintes aspetos: resumo, definição da problemática e objetivos, enquadramento teórico, metodologia, apresentação do artefacto e exposição dos resultados efetuados através de um questionário online.

O presente trabalho encontra-se dividido em três partes fundamentais que se complementam. A primeira parte consiste na definição do problema, dos objetivos e do tipo de metodologia utilizada, bem como a revisão bibliográfica. Na segunda parte é apresentado o website que foi realizado, para dar resposta ao problema identificado. Numa terceira parte foram apresentadas as devidas conclusões referentes ao questionário destinado a profissionais e estudantes da área das TI.

1.1. Definição do Problema e Objetivos

O desenvolvimento das tecnologias de RPA estão em constante evolução, necessitando de respostas céleres e eficazes por parte de quem as utiliza. Nem sempre existe esta resposta imediata.

Assim sendo, o objetivo geral do trabalho *Aplicação web – comunidade RPA* é criar interações de pessoas com interesses em comum e contribuir para a partilha da informação entre profissionais e aprendizes da área.

São também objetivos específicos deste estudo, aferir questões como: a segurança, arquitetura, requisitos de engenharia de *software* no desenvolvimento *web* e modelação de bases de dados.

Foi ainda intuito deste estudo, recolher opiniões sobre a pertinência dos objetivos propostos e do artefacto realizado.

1.2. Metodologia de Investigação *Design Science Research*

A abordagem metodológica *Design Science Research* deve produzir um artefacto viável na forma de um construto, modelo, método ou instanciação, com o objetivo de desenvolver soluções baseadas em tecnologias para problemas de negócios importantes e relevantes.

Deve-se utilizar métodos rigorosos na criação de artefactos, no sentido de resolver problemas que o investigador encontre.

A DSR preocupa-se com a qualidade e utilidade de um artefacto que deve ser apresentado por meio de métodos de avaliação bem definidos e executados (Hevner et al., 2004).

Conforme o autor Hevner et al.(2004), na tabela 1 estão apresentadas as sete diretrizes a serem abordadas para as pesquisas em *Design-Science*.

Tabela 1 - Diretrizes - Design-Science Research, adaptado de (Hevner et al., 2004)

Diretrizes - Design-Science Research	
Diretrizes	Descrição
Diretriz 1: Design do artefacto	A investigação de <i>design-Science</i> deve produzir um artefacto viável sob a forma de uma construção, um modelo, um método ou uma instanciação.
Diretriz 2: Relevância do Problema	O objetivo da investigação em <i>design-Science</i> é desenvolver soluções baseadas em tecnologia para problemas de negócio importantes e relevantes.
Diretriz 3: Avaliação de Design	A utilidade, qualidade e eficácia de um artefacto de conceção devem ser rigorosamente demonstradas através de métodos de avaliação bem executados.
Diretriz 4: Contribuições para a Investigação	Uma investigação eficaz em termos de <i>design-Science</i> deve fornecer contribuições claras e verificáveis nas áreas do artefacto de conceção, das fundações de conceção e/ou das metodologias de conceção.
Diretriz 5: Rigor de Investigação	A investigação de <i>design-Science</i> baseia-se na aplicação de métodos rigorosos tanto na construção como na avaliação do artefacto de design.
Diretriz 6: Design como um processo de pesquisa	A procura de um artefacto eficaz requer a utilização de meios disponíveis para alcançar os fins desejados, satisfazendo as leis no ambiente problemático.

Diretriz 7: Comunicação de Investigação	A investigação em termos de <i>design-Science</i> deve ser apresentada de forma eficaz tanto para os públicos orientados para a tecnologia como para os públicos orientados para a gestão.
---	--

2. Revisão da Literatura

2.1.RPA

A automação de processos robóticos (RPA) foi um termo dado por Pat Geary em 2012, chefe evangelista da *Blue Prism*. O termo refere-se a *robots* (bot) de *software* que permitem simular ações humanas, interagindo com sistemas digitais e de *software*. Os *robots* podem atuar em computação em nuvem ou através de *software* transmissível. Podem ainda executar um número variado de tarefas como: cortar e colar, envio de *emails*, trabalhar numa folha em Excel, efetuar *login* em aplicações, preencher e submeter formulários, consultar APIs entre outras. Ou seja, foca-se em tarefas que são repetitivas, libertando os trabalhadores para efetuarem outras atividades essenciais à empresa. O RPA pode ser usado num PC ou num sistema central, programando as ações de uma pessoa, criando a automatizando e em muitos casos, recorrendo à ajuda da inteligência artificial. O RPA permite a redução de custos e uma rápida e significativa recuperação do Retorno do investimento (ROI). Possibilita ainda uma maior rapidez na execução de processos e da sua entrega (Taulli, 2020).

2.1.1. Vantagens do RPA

Segundo o autor Taulli (2020) a vantagem que se destaca mais no RPA é o retorno do investimento (ROI). Conforme um inquérito feito a 250 empresas, relatório da *Economics Technology Trends* de 2019, onde muitas das empresas tinham receitas de 20 milhões de dólares. Cerca de 12% destas implementaram RPA, e metade tinha um ROI positivo no prazo de 18 meses.

O RPA é de fácil implementação, é de simples compreensão a nível da codificação, fornece um forte rasto em auditorias para melhor acompanhamento e correções. A nível de conformidade, uma vez que o contato com o humano é mínimo, a probabilidade de fraude diminui bastante. No atendimento ao cliente, as métricas de satisfação aumentam

significativamente, uma vez que os *robots* estão programados para trabalhar em grande escala e com menos erros.

2.1.2. Desvantagens do RPA

Embora o RPA seja uma boa solução e está a crescer exponencialmente, também tem as suas fraquezas, como:

- Custo de *software*;
- Custo de *robots* e máquinas onde os mesmos vão atuar;
- Custo da formação de recursos humanos e manutenção contínua;
- Sempre que as empresas efetuam atualizações nos processos ou nas aplicações, é necessária haver manutenção nos *robots*;
- À medida que os *robots* vão crescendo, pode-se tornar mais difícil a sua gestão, por isso é necessário haver uma forte colaboração das restantes TI.

2.2. Comunidades virtuais

Uma comunidade virtual é composta por um grupo de pessoas que usam a via *online* para se comunicarem, trocando ideias e interesses em comum.

O termo foi publicado pela primeira vez em 1984 através do projeto *Electronic Café*, que consistia na conexão de cinco restaurantes e um museu através de um link de vídeo.

O primeiro indício de comunidades virtuais teve início em 1968 por JCR Licklider e Robert Taylor através do projeto *ARPANET*. No início do século 21 expandiu-se e já faziam parte cerca de um bilião de pessoas com acesso à *internet*, onde realizavam discursos, socializavam, criavam negócios, serviços de redes sociais como o *Facebook*, comunidades de partilha de informação, de mídia entre outros (Rheingold, 2021).

2.3. Engenharia de Requisitos

A engenharia de requisitos é o processo da descoberta, análise, da documentação e verificação das funções e restrições de um sistema, onde se veem refletidas as necessidades de um cliente para um determinado sistema e o seu propósito (Sommerville, 2015).

Os requisitos são caracterizados por dois níveis, os de utilizador e os de sistema:

- Os requisitos de utilizador são uma declaração mais abstrata de alto nível, em linguagem natural e apoiado por diagramas. Deve indicar quais são os serviços a fornecer aos seus utilizadores e que restrições o sistema deverá operar.
- Mais detalhado, o requisito de sistema deve definir exatamente o que será implementado no software (funções, serviços e restrições operacionais).

Os requisitos do *software* geralmente são classificados como funcionais e não funcionais:

- Os requisitos funcionais estão diretamente ligados à funcionalidade de um sistema, onde descreve as funções que este deve executar. Esses requisitos dependem do tipo de *software* que se pretende desenvolver. Se forem caracterizados como requisitos de utilizador, devem ser escritos em linguagem natural para que os utilizador e gestores possam entender.
- Os requisitos não funcionais expressam as restrições sobre os serviços ou funções que um sistema deve ter. Podem estar relacionados às propriedades do sistema emergente, como o tempo de resposta, uso da memória, confiabilidade do sistema, entre outras. Estes requisitos surgem devido às necessidades do utilizador em consequência de restrições de orçamento, políticas organizacionais, necessidades de interagir com outro *software* ou *hardware* ou fatores externos, como regulamentos de segurança ou legislação de privacidade. Os requisitos não funcionais geralmente são mais críticos que os funcionais. Segundo Sommerville (2015) a falha de um requisito não funcional pode significar que todo o sistema fique inutilizável, Ex: se uma aeronave não refletir os requisitos de segurança e confiabilidade, não ira ser certificada como segura para poder operar.

2.4. Padrão de arquitetura de software MVC

O padrão MVC originalmente em 1979 foi designado por *Thing-Model-View-Editor* e posteriormente simplificado por MVC. Este é uma forma de separar regras e interesses no desenvolvimento de um *software*. O seu padrão é bastante utilizado, pois tem uma grande aplicabilidade no desenvolvimento de aplicações *web*. A sua arquitetura adiciona complexidade, no entanto a separação de regras de negócio através das suas camadas muito bem definidas, beneficia claramente a estrutura e *design* de uma aplicação. Proporciona ainda uma manutenção mais simples e um reaproveitamento de classes e

partes do projeto em trabalhos futuros. O MVC é usado em numerosas *frameworks* e encontra-se nas várias linguagens de programação, tais como: Java, C#, C ++, PHP, Windows, Mac (Jon Galloway *et al*, 2014).

Segundo Jon Galloway *et al* (2014) o MVC é separado em três camadas principais.

- *View* ou visão é a camada de apresentação que proporciona a entrada de dados e visualização de respostas geradas. Nas aplicações *web* é representado por HTML. A camada de apresentação deve refletir as alterações dos dados e estados da camada modelo.
- *Model* ou modelo é a camada que contém classes onde descrevem os dados e é responsável pelas regras de negócio. É nela que é realizado todas as operações de acesso ao SGBD, deixando os dados prontos para a camada do controlador que por sua vez renderiza e encaminha para a camada de apresentação.
- *Controller* ou controlador é o intermediário entre a camada de apresentação e a camada modelo. Tem como função controlar e coordenar o envio de requisições feitas pela camada visão. Define o comportamento da aplicação, interpreta as solicitações efetuadas pelos utilizadores e comunica-se com a camada de negócio que por sua vez devolve a informação e esta é renderizada na visão correta.

O autor indica que esta definição de MVC está ao nível da interface de utilizador. Não menciona como lidar com situações, tais como: questões da aplicação, acesso ao SGBD, interações de serviço, entre outras questões de desenvolvimento do software. Muitas destas questões estão definidas em *frameworks* que utilizam a arquitetura de *software* MVC.

2.5. Modelação de base de dados

A modelação de dados é o processo que permite demonstrar como serão construídas as estruturas de dados que dão suporte aos processos de negócios, através de uma representação visual. Regras e requisitos são definidos antecipadamente conforme *feedback* dos *stakeholders*. (IBM, 2021).

O processo de modelação é efetuado através da recolha de informação dos requisitos de negócios das partes interessadas e dos utilizadores finais, que serão traduzidas em estruturas de dados para formular um *design* concreto de base de dados.

Os modelos de dados dividem-se em três tipos:

- Modelos de dados conceituais: é a representação dos objetos que serão transformados em dados, utilizando uma visão simplificada. Geralmente são criados como parte do processo de recolha de requisitos do projeto. Nesta fase de modelação, o projeto é independente do SGBD. Modelos de dados conceituais incluem entidades, características e restrições, relacionamentos e os requisitos de segurança e integridade (Sommerville, 2015).

Figura 1 representa o modelo de dados conceituais.

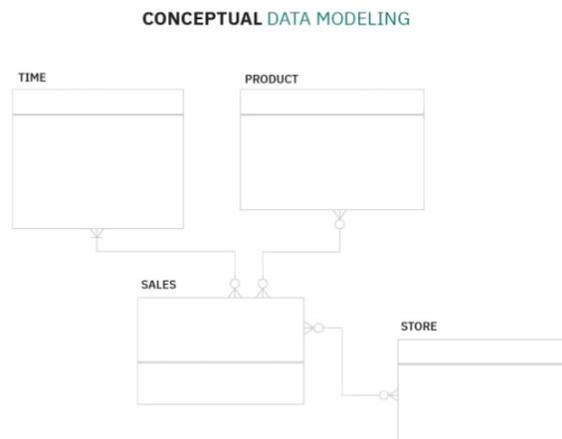


Figura 1 - Exemplo de modelos de dados conceituais (IBM, 2021)

- Modelos de dados lógicos: são menos abstratos, mais próximos do modelo físico, fornecem mais detalhes sobre os conceitos e relacionamentos de dados. Indicam atributos, tipos de dados e comprimento e mostram os relacionamentos entre as entidades. Neste nível o projeto ainda é independente do SGBD.

Figura 2 representa o modelo de dados lógicos.

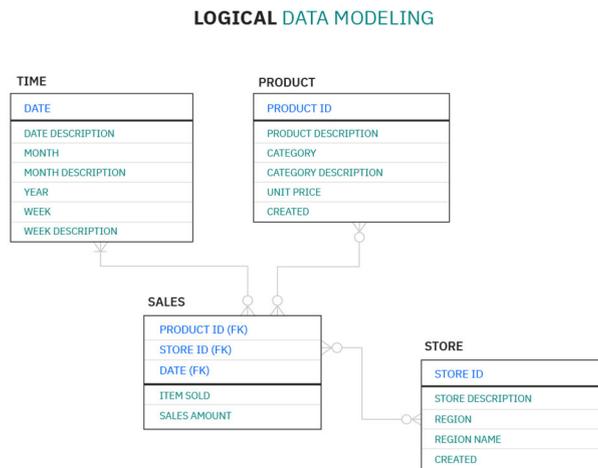


Figura 2 - Exemplo de modelos de dados lógicos (IBM, 2021)

Modelos de dados físicos: a partir do modelo lógico é efetuado o modelo físico. Fornecem um esquema de como os dados serão armazenados fisicamente em um SGBD. Estes detalham os componentes da estrutura física da base de dados, incluindo tabelas associativas, tipo de dados, chaves primárias e estrangeiras que são usadas para manter o relacionamento.

A **Figura 3** representa o modelo de dados físicos.

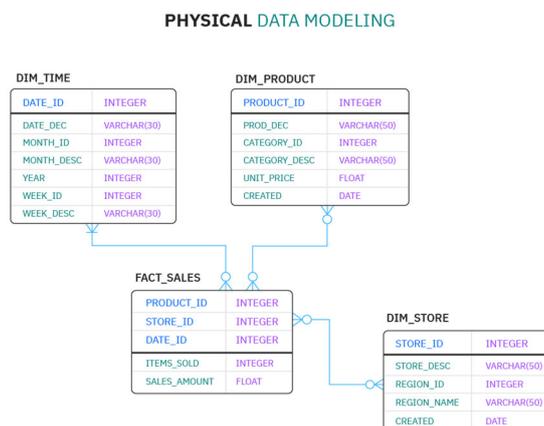


Figura 3 - Exemplo de modelos de dados físicos (IBM, 2021)

Tarefas para modelagem:

As tarefas devem ser realizadas de forma iterativa para que seja possível efetuar a modelagem de dados:

1. Identificar os tipos de entidade;
2. Identificar propriedades principais de cada entidade;
3. Identificar relacionamentos;
4. Mapear atributos para entidades;
5. Atribuir chaves conforme necessário, e normalizar para reduzir redundância;
6. Finalizar e validar o modelo de dados criado.

2.6.Arquitetura Monolítica

Segundo o autor Newman (2019), existem pelo menos três tipos de sistemas monolíticos:

- Arquitetura monolítica de processo único, quando um sistema não dividido é implementado como um único processo. Os diferentes componentes estão ligados a um programa dentro de uma única plataforma. Todo o código é implementado num único processo, podendo ter várias instâncias. Estes processos por ler e armazenarem informação num SGBD (Newman, 2019).

A **Figura 4** representa uma arquitetura monolítica de processo simples.

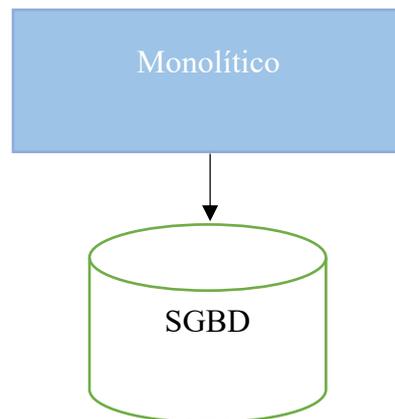


Figura 4 - Arquitetura monolítica de processo único, adaptado de (Newman, 2019)

- Arquitetura monolítica modular é uma variação do monolítico de processo único, com módulos separados onde cada um pode ser trabalhado separadamente. Permite um elevado grau de trabalho em paralelo, quando os módulos são bem definidos. Tendo em conta o autor Newman (2019), um dos desafios é a base dados que tende

a não ser decomposta. No entanto existem situações em que equipas efetuaram decomposição das mesmas.

A **Figura 5** representa uma arquitetura monolítica modular.

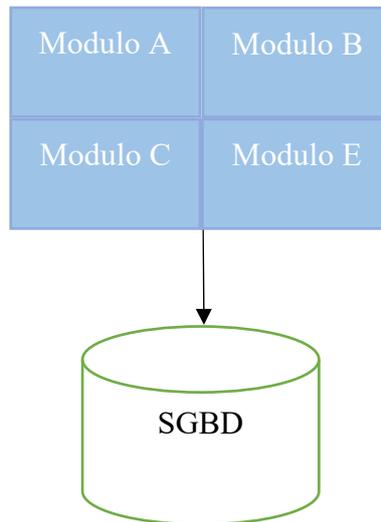


Figura 5 Arquitetura monolítica modular, adaptado de (Newman, 2019)

A **Figura 6** representa uma arquitetura monolítica modular com decomposição do SGBD.

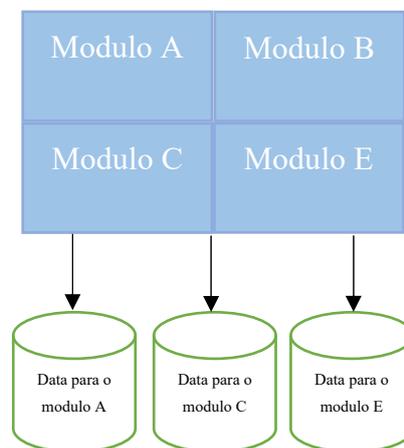


Figura 6 Arquitetura monolítica modular com decomposição do SGBD, adaptado de (Newman, 2019)

- Para o autor, Newman (2019) este sistema consiste em múltiplos serviços implementados em conjunto.

Vantagens da Arquitetura Monolítica:

- São simples de desenvolver, onde o sistema fica num único processo;
- São simples de testar, pois pode-se realizar testes à aplicação de ponta a ponta em um único local;
- A implementação no servidor é simples, pois é um processo único.
- Para projetos iniciais o custo é menor.

Desvantagens da Arquitetura Monolítica:

- Em relação à manutenção, quando a aplicação se torna cada vez maior o código é mais difícil de entender, crescendo o desafio de efetuar alterações rápidas;
- Quanto às alterações, é necessário efetuar uma nova implementação de toda aplicação;
- Quando a aplicação está em produção e é necessário efetuar alterações, um simples erro de código na aplicação pode levar a que esta fique inoperável;
- Uma vez escolhida a linguagem de programação, não é possível utilizar outra no projeto.

2.7.Desenvolvimento WEB

2.7.1. HTML

O HTML é uma linguagem de marcação baseada na SGML que foi desenvolvida em 1989 por Tim Berners-Lee. Esta teve o intuito de permitir que os investigadores pudessem partilhar grandes quantidades de dados. O HTML consiste em marcadores chamados de *tags*, que indicam a função e finalidade do texto.

No início de 1990 foram criados vários *browsers* de navegação *web* onde muitos expandiram a linguagem original, que levou a muitas incompatibilidades de visualização. Para colmatar estes problemas em 1994, Tim Berners-Lee formou a W3C para criar protocolos e diretrizes para garantir o crescimento a longo prazo da *Web*.

Atualmente a versão mais recente da linguagem de marcação de hipertexto é o HTML5 (Hoy, 2011).

2.7.2. Web Design

Web design refere-se á forma como os sites são apresentados na *internet*, tendo como principal objetivo a experiência dos utilizadores no desenvolvimento. O *design* web era focado para os desktops, no entanto a partir de 2010 o uso de navegadores móveis subiu de 2,86 para 7,02 por cento. As estatísticas mostravam que o uso do *Internet Explorer* tinha caído de 6,79 para 3,42 por cento, era importante dar foco ao design para navegadores móveis (Frain, 2012).

2.7.2.1. Web Responsive design

No *Web Responsive design*, o conteúdo move-se dinamicamente. Termo adotado por Eyhan Marcotte, consolidou três técnicas: *layout* flexível de grelha, imagens flexíveis e consultas de média. Segundo o autor Frain (2012) a abordagem para se ter no desenvolvimento é começar do mais pequeno *viewport* e de seguida melhorar e adotar para conteúdos de vistas maiores.

2.7.2.2. Adaptive web design

No *Adaptive web design*, o conteúdo é fixado em tamanhos, que corresponde à dimensão dos dispositivos. Este deteta o dispositivo do utilizador e adapta o *layout* adequado (Delaney, 2020).

2.7.3. PHP

PHP é uma linguagem de programação de *script*, de código aberto, para desenvolvimento de aplicações e páginas *web*. Pode ser incorporada nas páginas HTML para gerar conteúdo dinâmico sempre que o utilizador solicitar. É uma linguagem interpretada no lado do servidor. Concebida em 1994 por Rasmus Lerdof, foi adotada por várias pessoas e sofreu constantes atualizações para se tornar num produto amplo e maduro (Welling & Thomson, 2016).

Atualmente o PHP foi atualizado para a versão 8.0, lançada em 2020, conta com novos recursos e atualizações (PHP, 2021).

2.7.3.1. Composer

Composer é um gestor de dependências desenvolvido para PHP. Este permite que seja efetuado a declaração de bibliotecas realizando a sua gestão, ou seja instala-as quando o utilizador as solicita e efetua as atualizações sempre que estas sejam alvo de *updates* (*Composer, 2019*).

2.8. Segurança com PHP

A segurança de aplicações *web* é um componente essencial no seu desenvolvimento. Com a evolução das TI, surgem novas formas de ataques. O risco de dados sensíveis estarem expostos é alto, pelo que é necessário ter uma especial atenção à segurança, aquando do desenvolvimento das aplicações *web* (Medeiros, 2016)

2.8.1. Injeção de SQL

A injeção de SQL é um ataque formado com a execução de instruções de SQL maliciosas, podendo expor, corromper, eliminar, alterar dados do SGBD, entre outros. Este tipo de vulnerabilidade já teve impacto em várias empresas mundiais como: o Yahoo que em 2012 teve 450 000 mil *passwords* expostas de utilizadores. No mesmo ano, a Nvidia teve 400 000 mil *passwords* comprometidas, a Adobe também fez parte deste leque, tendo 150 000 mil *passwords* também comprometidas (Edmunds, 2016).

A injeção de SQL funciona quando o programador passa diretamente para a *query* de SQL os *inputs*, segundo Edmunds (2016), “1 MySQL *_query*('UPDATE users, 2 SET *first_name*=' . \$_POST['*first_name*'] . "' 3 WHERE id=1001')“ (p.2).

Para o mesmo autor, um utilizador mal intencionado pode realizar operações como, “UPDATE users SET *first_name*="Liz", *last_name*="Lemon"; --" WHERE id=1001”; desta forma altera todos os utilizadores com *first_name* para Liz e *last_name* para Lemon.

Para evitar a injeção de SQL deve-se usar a *class* PDO, do PHP e o método *prepare* com o *bindParam*. Desta forma, todos os parâmetros serão validados, segundo Edmunds (2016), “ 1 \$db = new PDO(...); 2 \$query = \$db->prepare('UPDATE users 3 SET *first_name* = :*first_name* 4 WHERE id = :*id*'); 5 6 \$query->execute([7 ':*id*' => 1001, 8 ':*first_name*' => \$_POST['*first_name*'] 9]); ” (p.3).

As *queries* parametrizadas e as declarações preparadas são as melhores técnicas a utilizar para se evitar a injeção de SQL. Outra forma de proteção pode ser o uso dos procedimentos armazenados (SP), que são funções que estão diretamente no SGBD. Existe menos probabilidade de injeção de SQL com as SP, no entanto segundo o autor Edmunds (2016), as SP são difíceis de testar, difíceis de controlar a versão, uma vez que estas estão no SGBD e não no código da aplicação. Este processo pode limitar o número de pessoas que estão aptas a modificar e atualizar as SP.

2.8.2. JavaScript

O JavaScript não deve ser usado na validação final dos dados. Pode ser usado para validação no lado do cliente, uma vez que oferece resposta assíncrona. No entanto deve-se sempre validar os dados em *backend* (Edmunds, 2016).

2.8.3. Hash passwords

Hash é uma forma de representar um determinado dado com uma *string* única, contendo uma cadeia de caracteres. Para as *passwords*, o hash deve ser unidirecional, desta forma é possível descriptar uma *password* através da original. Existem vários tipos de algoritmos de *hash* como:

- MD5 - Foi muito usado, uma vez que é suportado por diversos SGBD. No entanto está matematicamente provado que não é confiável. Em 2005 investigadores provaram que o uso desta técnica pode facilmente gerar colisões; uma colisão é quando duas passwords diferentes produzem o mesmo hash.
- SHA-1 - Também em 2005 investigadores provaram que este tipo de hash é inseguro, devido à sua alta probabilidade de colisão. Assim como o MD5 não é aconselhável para uso de criação de *passwords*.
- SHA-256 / SHA-512, SHA-256 - usa 32 bits enquanto a SHA-512 usa 64 bits, o número de rondas também difere, no entanto, o algoritmo é idêntico. Atualmente são consideradas seguras.
- BCrypt - lançado em 1999, baseado na cifra do *Blowfish*. Incorpora um *salt* (cadeia de caracteres aleatórios) para proteção contra ataques. O custo pode ser aumentado, tornando-se mais segura, no entanto o processo de verificação da

password é mais lento. É considerada segura, possui uma limitação de 72 caracteres para *passwords* únicas.

2.8.4. MYSQL

MySQL é um Sistema de Gestão de Base de Dados relacional bastante rápido, disponível publicamente desde 1996, sendo um dos mais populares a nível mundial. O MySQL permite armazenar, pesquisar, classificar e recuperar os dados de forma eficaz. Através do seu servidor, permite controlar os acessos para que possam trabalhar vários utilizadores em simultâneo. O MySQL utiliza a linguagem de consulta de dados (SQL) (Welling & Thomson, 2016).

Possui as seguintes funcionalidades:

- Desenvolvimento SQL - permite criar e efetuar gestão de conexões com servidores de base de dados;
- Modelagem de dados - permite criar modelos graficamente, efetuar engenharia reversa e direta entre esquemas e base de dados. O editor permite criar e editar rapidamente tabelas, procedimentos, gatilhos, particionamento, rotinas e visualizações;
- Administração do servidor - é possível administrar e criar instâncias do servidor, realizar *backups*, auditorias, desempenho e verificar a integridade dos dados;
- Migração de dados - permite migrar dados de outro SGBD, como Microsoft SQL *Server e Access, SQLite, SQL Anywhere, PostgreSQL* e outras tabelas *RDBMS*, objetos e dados para MySQL (MySQL, 2020).

2.8.5. Bibliotecas

2.8.5.1. AJAX

AJAX, acrónimo de *Asynchronous JavaScript and XML*, é uma técnica de desenvolvimento de aplicações *web*, ou seja permite efetuar chamadas ao servidor através de *JavaScript* e atualizar partes da página HTML sem necessitar de recarregar a página por completo (Darie & Brinzarea, 2009).

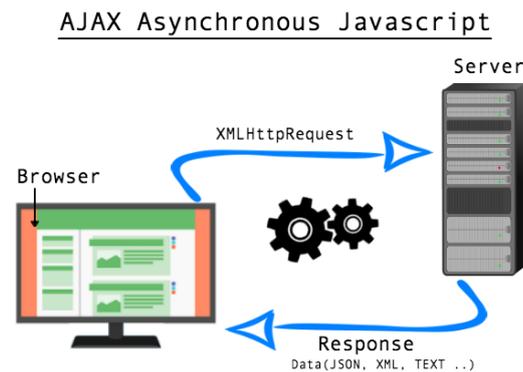


Figura 6 - Exemplo do funcionamento de AJAX (Social Network, sem data)

2.8.5.2. jQuery

O *jQuery* é uma biblioteca para *JavaScript*, *opensource* que foi criada em 2006 por *John Resig*. Foi projetada para simplificar a programação do *JavaScript*, como pedidos Ajax e manipulação da DOM. Com o lema "Escreva menos, faça mais.", o *jQuery* compacta várias linhas de código em uma única função, reduzindo a reescrita de todos os blocos repetidamente (Bibeault et al., 2015)

2.8.5.3. Materialize

Materialize é uma linguagem que utiliza o conceito *web responsive*, direcionada para a estilização de páginas *web*. Criada pela google com o objetivo de fornecer ao utilizador uma experiência unificada em todos os seus produtos e dispositivos (About - *Materialize*, 2021).

2.8.5.4. Trumbowyg

O *Trumbowyg* é um editor *WYSIWYG* acrónimo de (*What You See Is What You Mean*) que permite ao utilizador criar publicações *web* e visualizar o resultado final em tempo real. O utilizador não necessita de qualquer conhecimento de HTML, pois o editor gera o conteúdo (Khalili et al., 2012).

2.8.5.5. API RESTFull Countries

A *API RESTFull Countries* é de código aberto e gratuito que devolve informação sobre os países através da arquitetura de *software RESTFull* (*Alejandro Matos / Rest Countries*, 2021)

2.8.5.6. PHPMailer

O *PHPMailer* é uma biblioteca de código aberto, para enviar *emails* a partir do PHP (*PHPMailer*, 2011/2021)

Análise de requisitos

O levantamento dos requisitos da *aplicação web – Comunidade RPA* foi baseado em aplicações similares, do mesmo âmbito.

3. Requisitos funcionais da aplicação web

Tabela 2 - Requisitos funcionais

ID	Requisito
RF01	Deve permitir ao visitante criar um utilizador de acesso.
RF02	Deve permitir efetuar <i>login e logout</i> ao utilizador registado.
RF03	Deve permitir ao utilizador registado, efetuar operações de CRUD ao perfil.
RF04	Deve permitir ao utilizador registado, efetuar alteração de password.
RF05	Deve permitir ao utilizador registado, excluir a sua conta.
RF06	Deve permitir o utilizador efetuar operações de CRUD aos tópicos criados pelo próprio, mencionando a quantidade de edições.
RF07	Deve permitir ao utilizador registado, responder aos tópicos e a repostas dos mesmos.
RF08	Deve permitir ao utilizador registado efetuar <i>like</i> a uma resposta.
RF09	Deve informar a quantidade de tópicos recentes
RF10	Deve permitir pesquisa de tópicos

RF11	Deve permitir informação: quando o tópico foi criado, por quem, quantidade de visitas e quantidade de respostas.
RF12	Deve permitir navegar por categorias e subcategorias.
RF13	Deve informar o nome de quem criou a resposta, quantidade de edições e data de criação
RF14	Deve ter níveis de <i>users</i> como: proprietário, moderador e utilizador.
RF15	O proprietário deve poder gerir toda a aplicação através de ações como: editar informações da comunidade, mover, moderar e aprovar postagens, adicionar outros membros e eliminar a comunidade.
FR16	O Moderador deve poder efetuar funções como: mover, moderar e aprovar postagens e editar informações sobre a comunidade.

3.1.Requisitos não funcionais

3.1.1. Segurança

Tabela 3 - Requisitos não funcionais de Segurança

ID	Requisito
RNF01	Deve criptografar as <i>passwords</i> do utilizador.
RNF02	Deve utilizar o protocolo <i>https</i> .

3.1.2. Usabilidade

Tabela 4 - Requisitos não funcionais de usabilidade

ID	Requisito
RNF03	A aplicação <i>web</i> deve ser <i>web responsive</i> .

3.1.3. Restrições

Tabela 5 - Requisitos não funcionais de restrições

ID	Requisito
RNF05	A aplicação <i>web</i> deverá ser entregue na data definida pela universidade.
RNF06	A aplicação deve utilizar a arquitetura monolítica.

3.1.4. Manutenibilidade

Tabela 6 - Requisitos não funcionais de manutenibilidade

ID	Requisito
RFF07	A aplicação <i>web</i> deverá ser contruída na arquitetura <i>Model, View e Controller</i> (MVC)

3.1.5. Portabilidade e Interoperabilidade

Tabela 7 - Requisitos não funcionais de Portabilidade e Interoperabilidade

ID	Requisito
RNF08	A aplicação <i>web</i> deverá utilizar o sistema de gestão de base de dados MySQL .
RNF09	O desenvolvimento da aplicação será efetuado no sistema operativo Windows através do editor <i>Visual Studio Code</i> .
RNF10	A linguagem de programação empregue no desenvolvimento deverá ser o PHP versão 8.1.
RNF11	As ferramentas utilizadas no desenvolvimento da aplicação <i>web</i> deverão ser de código aberto.

3.1.6. Desempenho

Tabela 8 - Requisitos não funcionais de desempenho

ID	Requisito
RNF12	No sentido de diminuir o tempo de resposta, a aplicação <i>web</i> deverá ser desenvolvida recorrendo à linguagem de programação <i>JavaScript</i> e à técnica <i>Asynchronous JavaScript And XML (Ajax)</i> .

3.2. Em seguimento aos requisitos apresentados nas tabelas 1, 2, 3, 4, 5, 6 e 7, foram efetuados os diagramas de caso de uso.

A Figura 7 representa o diagrama de casos de uso.

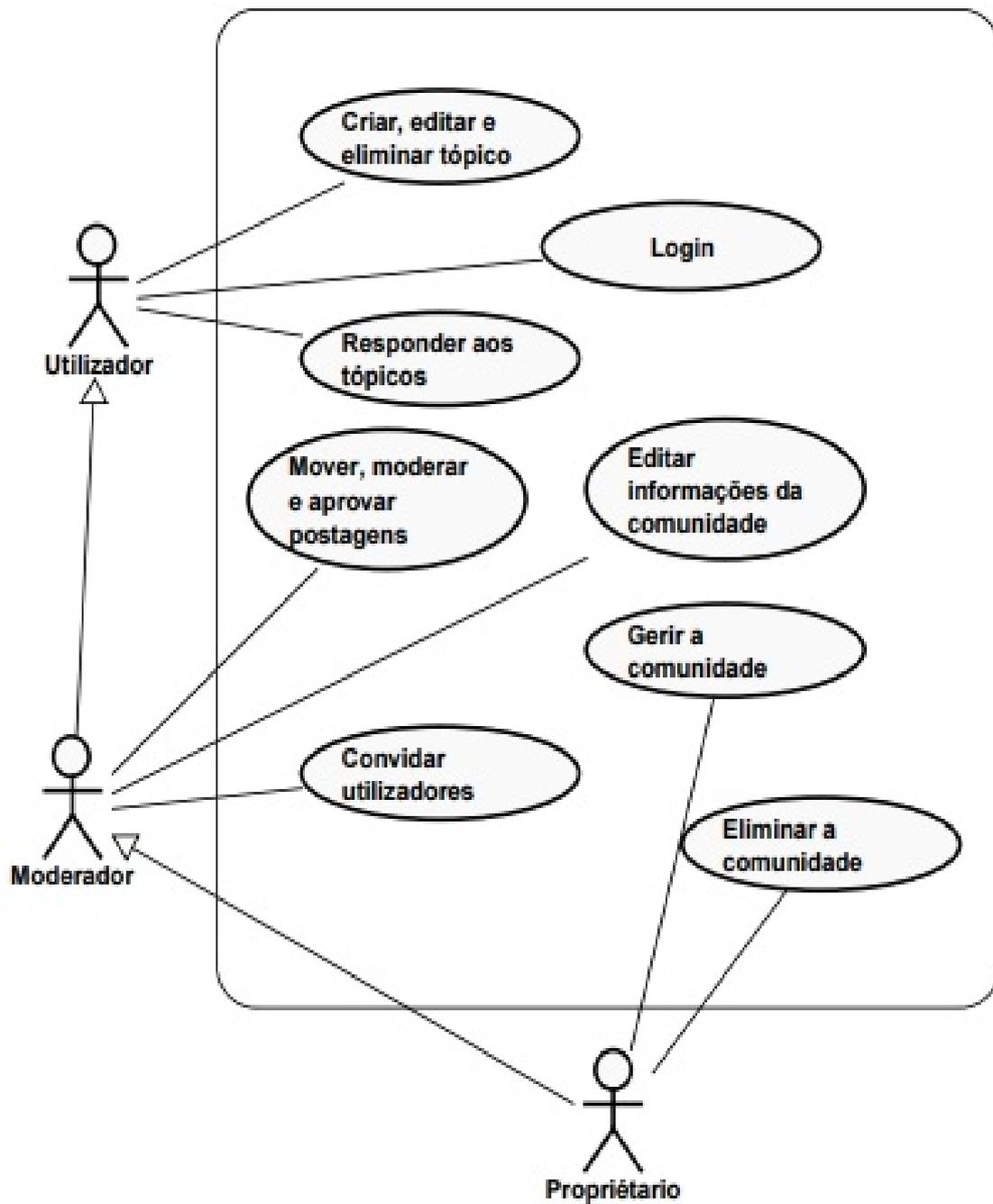


Figura 7 - Diagrama de caso de uso

4. Desenvolvimento da aplicação

Para o desenvolvimento da aplicação foi utilizado o sistema monolítico referenciado em 2.6. Foi optado por este sistema por ser de fácil implementação, de baixo custo para projetos iniciais, a implementação em um servidor também é simples pois é um processo único. A estrutura do projeto foi optada pela arquitetura padrão *Model View e Controller*, apresentado em 2.4. Esta escolha deve-se às vantagens que esta arquitetura traz no desenvolvimento deste tipo de aplicações, pois faz uma separação clara das camadas de visualização e de negócio. Neste tipo de arquitetura, e uma vez que o projeto está bem definido, a organização e manutenção do sistema torna-se mais fácil, bem como o reaproveitamento de código, principalmente na camada de modelo.

A linguagem de programação utilizada foi PHP, apresentada em 2.7.3, com o gestor de dependências *Composer* referenciado em 2.7.3.1. A linguagem de programação utilizada no desenvolvimento dinâmico do *frontend* foi JavaScript referenciada em 2.8.2.

As bibliotecas utilizadas, foram: *PHPMailer* para envio de *emails*, apresentado em 2.8.5.6; O *jQuery* para realizar todas as funções em *JavaScript* e requisições AJAX com JSON, HTML5 para as páginas web; Materialize e CSS3 para efetuar a estilização de todas as páginas de html e como editor *WYSIWYG* foi utilizada a biblioteca *Trumbowyg* para o utilizador efetuar *posts* apresentado em 2.8.5.4.

Foi escolhido o *design web responsive*, para desenvolvimento da aplicação *web* em *frontend* referenciado no ponto 2.7.2.1.

Como SGBD foi utilizado o MYSQL, exposto em 2.8.4.

4.1. Base de dados

Para a base de dados foi optado pelo modelo de entidade relacional estendido (EER).

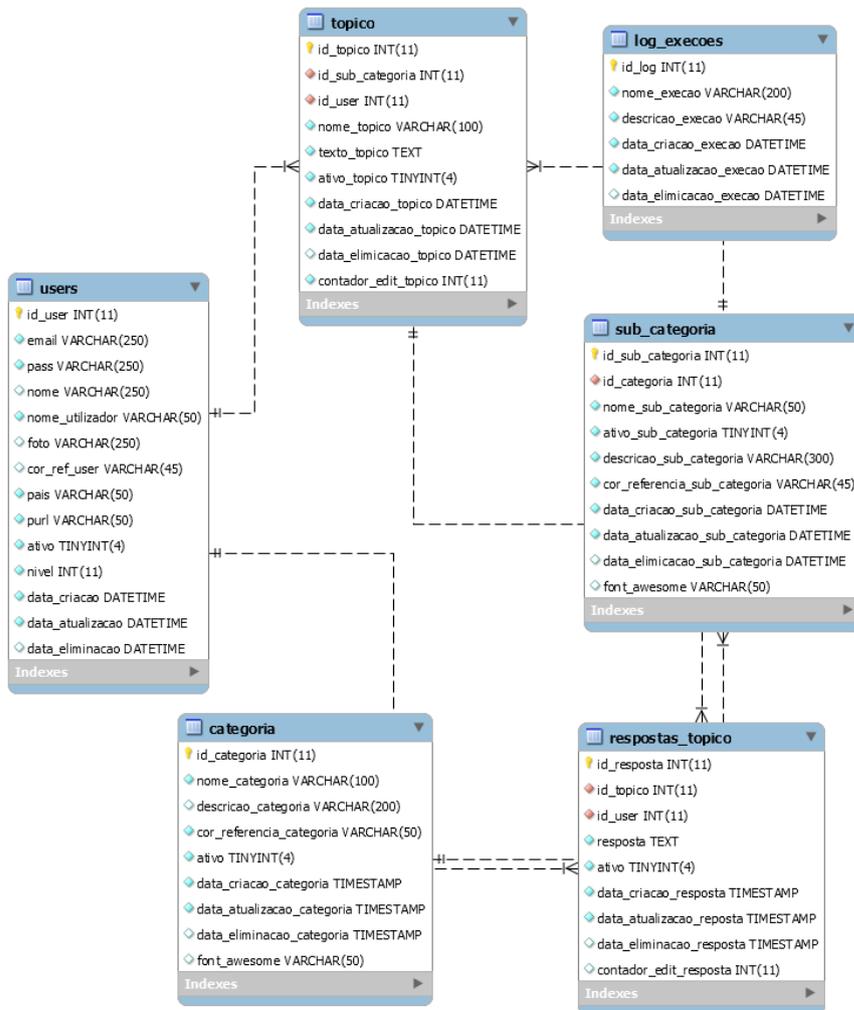


Figura 8 – Modelo EER da base de dados

4.2. Estrutura das tabelas

4.2.1. Tabela users

Tabela 9 – Campo e atributos da tabela users

Campo	Atributos
id_user	Chave primária, campo não nulo de auto incremento e tipo inteiro

email	Campo não nulo do tipo <i>varchar</i>
pass	Campo não nulo do tipo <i>varchar</i>
nome	Campo nulo do tipo <i>varchar</i>
nome_utilizador	Campo não nulo, do tipo <i>varchar</i>
foto	Campo que pode ser nulo, do tipo <i>varchar</i>
cor_ref_user	Campo não nulo, do tipo <i>varchar</i>
pais	Campo não nulo, do tipo <i>varchar</i>
purl	Campo não nulo, do tipo <i>TinyInt</i>
ativo	Campo não nulo, do tipo <i>int</i>
data_criacao	Campo não nulo do tipo <i>TimeStamp</i> , com valor por defeito <i>CURRENT_TIMESTAMP()</i>
data_atualizacao	Campo não nulo do tipo <i>TimeStamp</i> , com valor por defeito <i>CURRENT_TIMESTAMP()</i> e ao efetuar <i>update</i> á tabela <i>CURRENT_TIMESTAMP()</i>
data Eliminacao	Campo nulo do tipo <i>TimeStamp</i>

4.2.2. Tabela categoria

Tabela 10 – Campo e atributos da tabela categoria

Campo	Atributos
id_categoria	Chave primária, campo não nulo de auto incremento, e tipo inteiro;
nome_categoria	Campo não nulo do tipo <i>varchar</i>
atualizaç_categoria	Campo não nulo, do tipo <i>varchar</i>
cor_referencia_categoria	Campo não nulo, do tipo <i>varchar</i>
ativo	Campo não nulo, do tipo <i>TinyInt</i> ;
data_criacao_categoria	Campo não nulo do tipo <i>TimeStamp</i> , com valor por defeito <i>CURRENT_TIMESTAMP()</i>
data_atualizacao_categoria	Campo não nulo do tipo <i>TimeStamp</i> , com valor por defeito <i>CURRENT_TIMESTAMP()</i> e ao efetuar <i>update</i> á tabela <i>CURRENT_TIMESTAMP()</i>
data Eliminacao_categoria	Campo nulo do tipo <i>TimeStamp</i>
font_awesome	Campo não nulo, do tipo <i>varchar</i>

4.2.3. Tabela sub_categoria

Tabela 11 – Campo e atributos da tabela sub_categoria

Campo	Atributos
id_sub_categoria	Chave primária, campo não nulo de auto incremento, e tipo inteiro;

id_categoria	Chave estrangeira, campo não nulo, e tipo inteiro
nome_sub_categoria	Campo não nulo, do tipo <i>varchar</i>
ativo_sub_categoria	Campo não nulo, do tipo <i>tinyint</i>
38tualizaç_sub_categoria	Campo não nulo, do tipo <i>varchar</i>
cor_referencia_sub_categoria	Campo não nulo, do tipo <i>varchar</i>
data_criacao_sub_categoria	Campo não nulo do tipo <i>TimeStamp</i> , com valor por defeito <i>CURRENT_TIMESTAMP()</i>
data_atualizacao_sub_categoria	Campo não nulo do tipo <i>TimeStamp</i> , com valor por defeito <i>CURRENT_TIMESTAMP()</i> e ao efetuar <i>update</i> á tabela <i>CURRENT_TIMESTAMP()</i> .
Data_elimicacao_sub_categoria	Campo nulo do tipo <i>TimeStamp</i> ,
font_awesome	Campo não nulo, do tipo <i>varchar</i> .
Restrição	Chave estrangeira id_categoria referencia o campo id_categoria da tabela categoria.

4.2.4. Tabela respostas_topico

Tabela 12 – Campo e atributos da tabela respostas_topico

Campo	Atributos
id_resposta	Chave primária, campo não nulo de auto incremento e tipo inteiro.
Id_topico	Chave estrangeira, campo não nulo e tipo inteiro.
Id_user	Chave estrangeira, campo não nulo e tipo inteiro.
Resposta	Campo não nulo do tipo 38tual.
Ativo	Campo não nulo, do tipo <i>tinyint</i>
data_criacao_resposta	Campo não nulo do tipo <i>TimeStamp</i> , com valor por defeito <i>CURRENT_TIMESTAMP()</i> .
Data_atualizacao_reposta	Campo não nulo do tipo <i>TimeStamp</i> , com valor por defeito <i>CURRENT_TIMESTAMP()</i> e ao efetuar <i>update</i> á tabela <i>CURRENT_TIMESTAMP()</i> .
Data Eliminacao_resposta	Campo nulo do tipo <i>TimeStamp</i> ,
contador_edit_resposta	Campo não nulo, do tipo inteiro e com valor por defeito 0.
Restrição	Chave estrangeira id_topico referencia o campo id_topico da tabela tópico
Restrição	Chave estrangeira id_user referencia o campo id_user da tabela users.

4.2.5. Tabela tópico

Tabela 13 – Campo e atributos da tabela tópico

Campo	Atributos
id_topico	Chave primária, campo não nulo de auto incremento e tipo inteiro.
Id_sub_categoria	Chave estrangeira, campo não nulo e tipo inteiro.
Id_user	Chave estrangeira, campo não nulo e tipo inteiro.
Nome_topico	Campo não nulo do tipo <i>varchar</i> .
Texto_topico	Campo não nulo do tipo <i>Text</i> .
Ativo_topico	Campo não nulo, do tipo <i>TinyInt</i> .
Data_criacao_topico	Campo não nulo do tipo <i>TimeStamp</i> , com valor por defeito <i>CURRENT_TIMESTAMP()</i> .
Data_atualizacao_topico	Campo não nulo do tipo <i>TimeStamp</i> , com valor por defeito <i>CURRENT_TIMESTAMP()</i> e ao efetuar <i>update</i> á tabela <i>CURRENT_TIMESTAMP()</i> .
Data_elimicacao_topico	Campo nulo do tipo <i>TimeStamp</i> .
Contador_edit_topico	Campo não nulo, do tipo inteiro e com valor por defeito 0.
Restrição	Chave estrangeira id_sub_categoria referencia o campo id_sub_categoria da tabela sub_categoria.
Restrição	Chave estrangeira id_user referencia o campo id_user da tabela users.

4.2.6. Tabela exceções

Tabela 14 – Campo e atributos da tabela exceções

Campo	Atributos
id_log	Chave primária, campo não nulo de auto incremento e tipo inteiro.
Nome_execao	Campo não nulo do tipo <i>varchar</i> .
Descricao_execao	Campo não nulo do tipo <i>varchar</i> .
Data_criacao_execao	Campo não nulo do tipo <i>TimeStamp</i> , com valor por defeito <i>CURRENT_TIMESTAMP()</i> .
Data_atualizacao_execao	Campo não nulo do tipo <i>TimeStamp</i> , com valor por defeito <i>CURRENT_TIMESTAMP()</i> e ao efetuar <i>update</i> á tabela <i>CURRENT_TIMESTAMP()</i> .
Data_elimicacao_execao	Campo nulo do tipo <i>TimeStamp</i> .

4.3.Desenvolvimento prático da aplicação

Na **Figura 2** **Figura 9** é identificado a estrutura do projeto seguindo a arquitetura padrão MVC.

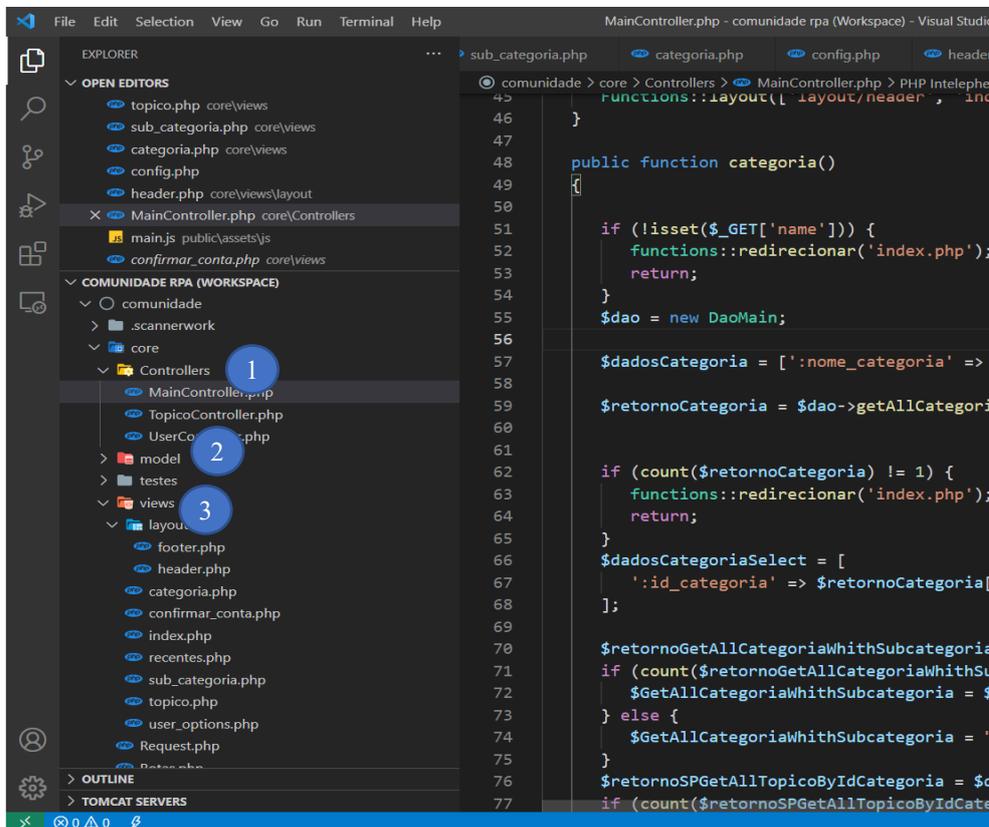


Figura 9 - Estrutura do desenvolvimento do projeto em MVC

Os pontos marcados na **Figura 9**, apresentam:

1. Neste ponto é identificado a camada *Controller*, onde vão ficar todos os controladores.
2. Neste ponto é identificado a camada *Model*, onde vão ficar todos os modelos.
3. Neste ponto é identificado a camada *View*, onde se encontra as *views*.

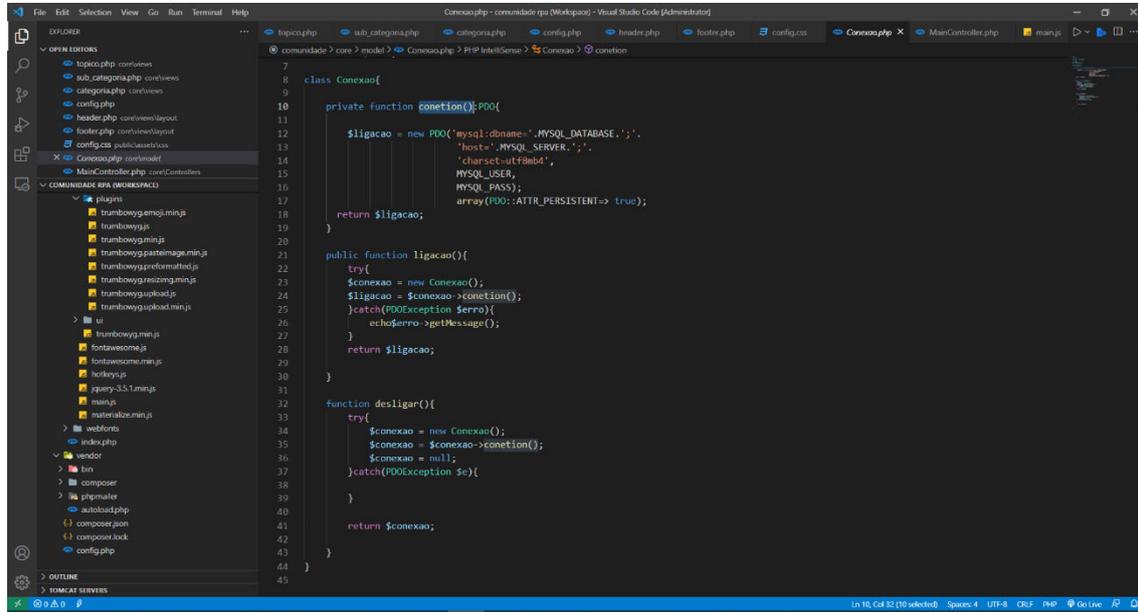
4.4.Métodos de acesso ao SGBD.

4.4.1. Conexão com o SGBD (MySQL)

A **Figura 10** apresenta um método privado com o nome *connetion()*, para efetuar a conexão com o SGBD através da classe PDO. Esta recebe como parâmetros os dados relativos à

ligação, estando declarados em variáveis globais num ficheiro de configuração (*config.php*).

A figura indica ainda os métodos de ligar e desligar com o SGBD.



```
7 class Conexao{
8
9
10 private function conetion():PDO{
11
12     $ligacao = new PDO("mysql:dbname='MYSQL_DATABASE';",
13                       "host='MYSQL_SERVER';",
14                       "charset=utf8mb4",
15                       MYSQL_USER,
16                       MYSQL_PASS);
17     array(PDO::ATTR_PERSISTENT => true);
18
19     return $ligacao;
20 }
21
22 public function ligacao(){
23     try{
24         $conexao = new Conexao();
25         $ligacao = $conexao->conetion();
26     }catch(PDOException $erro){
27         echo$erro->getMessage();
28     }
29     return $ligacao;
30 }
31
32 function desligar(){
33     try{
34         $conexao = new Conexao();
35         $conexao = $conexao->conetion();
36         $conexao = null;
37     }catch(PDOException $e){
38
39     }
40
41     return $conexao;
42 }
43
44 }
45 }
```

Figura 10 - Ligação com o SGBD e métodos para ligar e desligar da BD

4.5.Sistema de rotas

A Figura 11 representa o sistema de rotas através do método *get*.

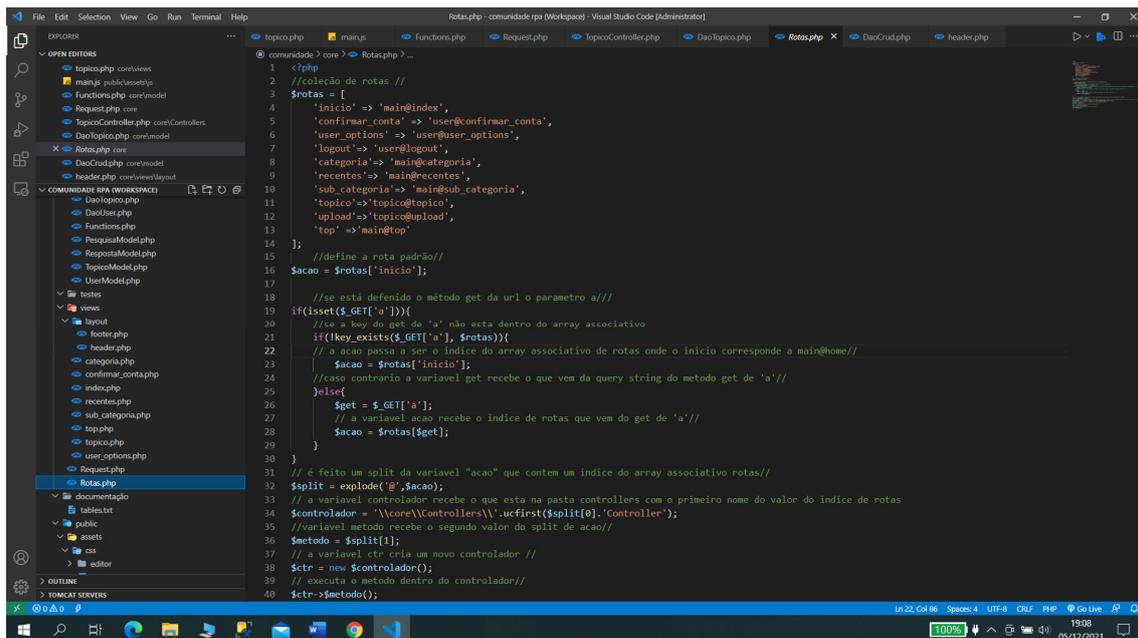
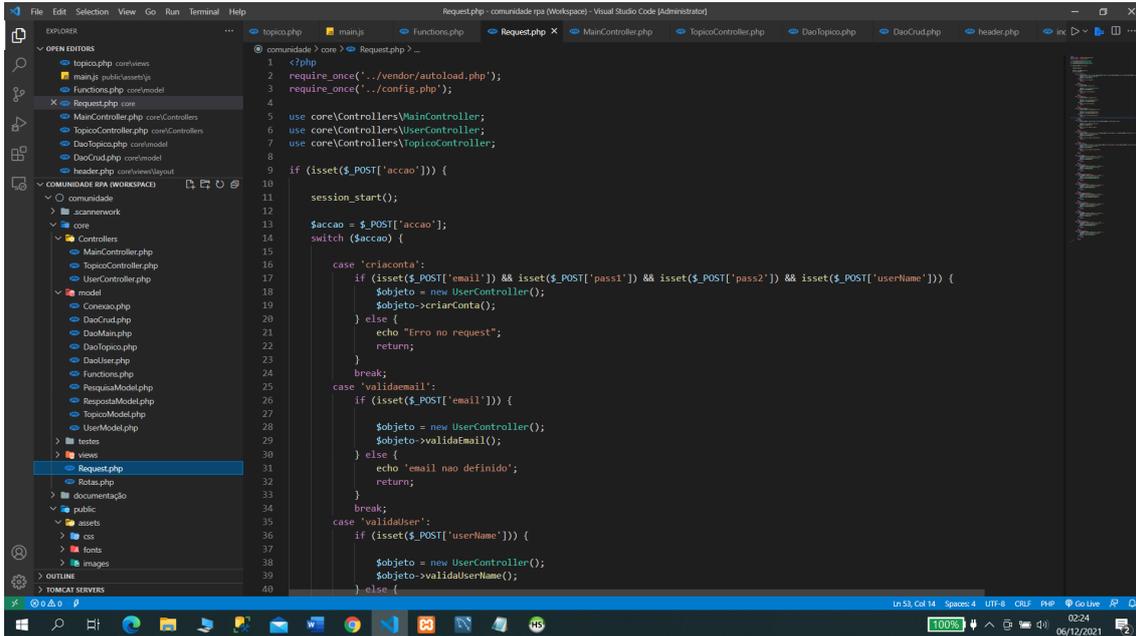


Figura 11 - Sistema de rotas

Para o sistema de rotas é criado um *array* associativo onde o índice é o nome da rota e o valor de cada índice vai representar o *Controller* e o método respetivo, ou seja, é efetuado um *split* ao valor através da arroba (@), onde a primeira posição do *array* (0) será o *Controller* e a segunda posição (1) será o método.

É verificado também se existe no método get, com o parâmetro (a) o índice das rotas, caso exista, cria uma instância do respetivo Controller e executa o método correspondente.

4.6. Tratamento de *request* via JavaScript



4.7. Criação e confirmação de *hash*

A Figura 12 representa o método para criação de um utilizador, onde é efetuado também a criação de um *hash* para *password*.

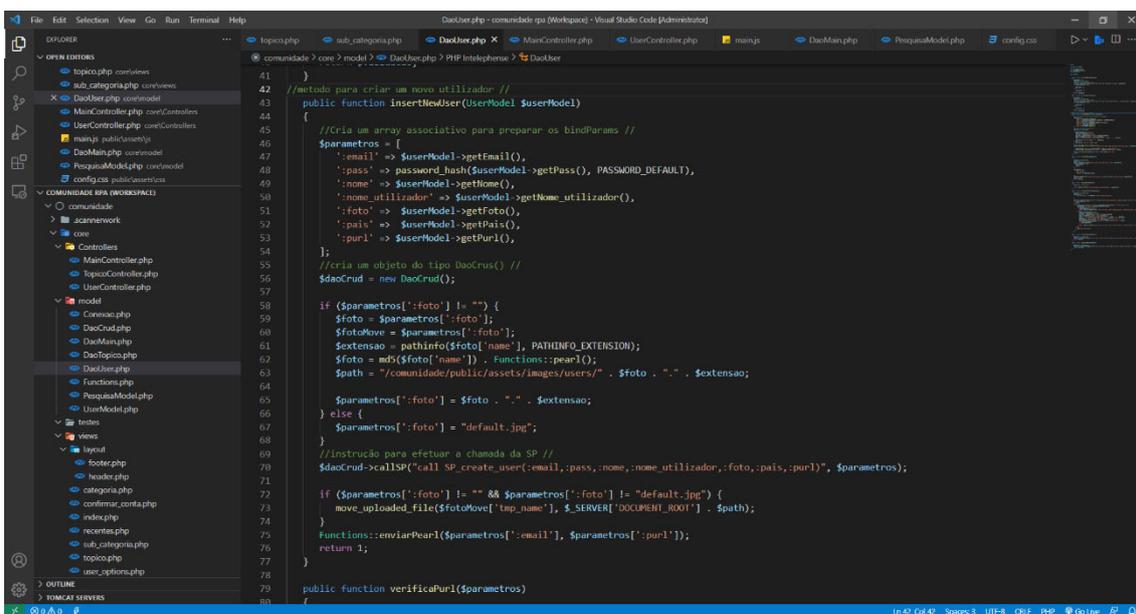
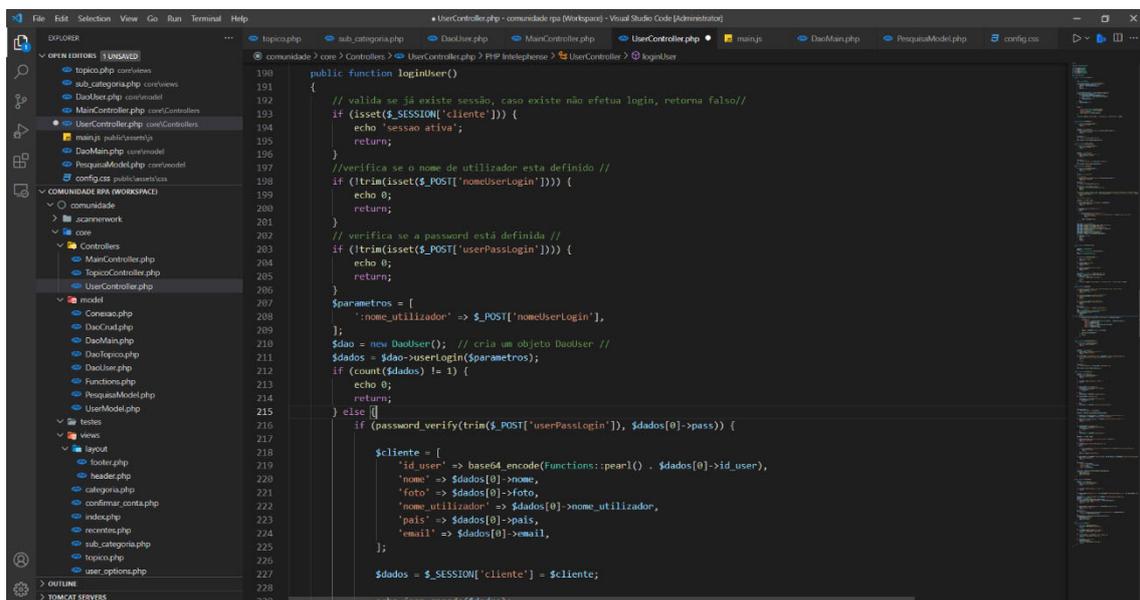


Figura 12 - Método para criação de utilizador, com *hash* da *password*

Conforme apresentado na **Figura 12**, foi criado um método com o nome *insertNewUser()* que recebe como parâmetro um modelo (class – *UserModel.php*) de um *user*. É efetuado a criação de um *hash* para a *password* do utilizador, através do método *password_hash(input, tipo_de_encriptação)*, utilizando o parâmetro *PASSWORD_DEFAULT* que representa o algoritmo *bcrypt* conforme referenciado em 2.8.3.

Na **Figura 13** foi criado um método para efetuar login, onde também é efetuado a verificação do *hash*.

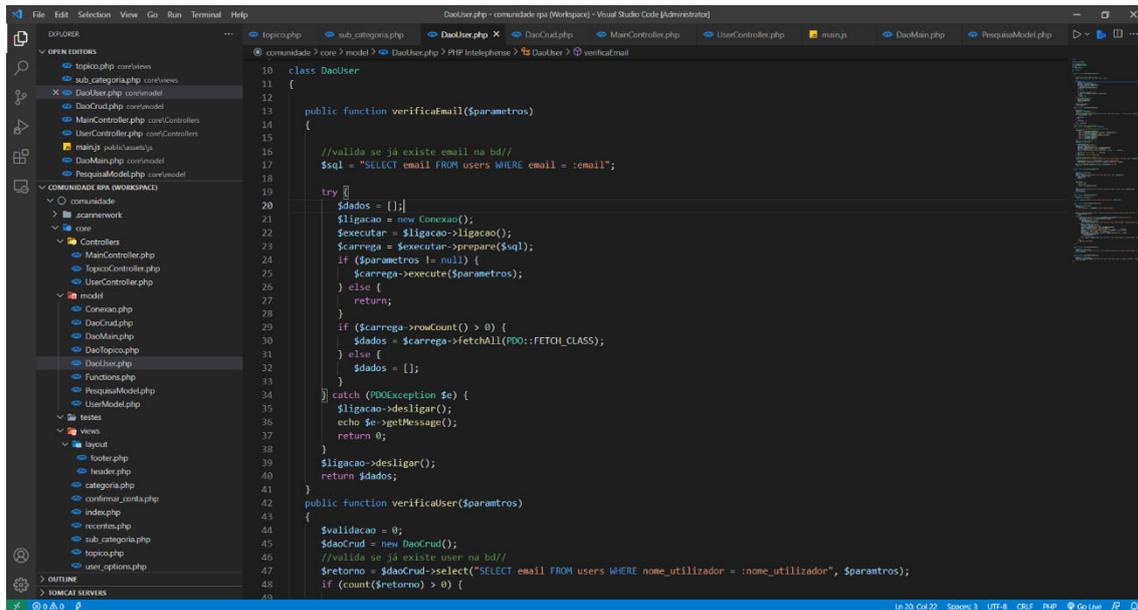


```
public function loginUser()
{
    // valida se já existe sessão, caso existe não efetua login, retorna falso//
    if (isset($_SESSION['cliente'])) {
        echo "sessao ativa";
        return;
    }
    //verifica se o nome de utilizador esta definido //
    if (trim(isset($_POST['nomeUserLogin']))){
        echo 0;
        return;
    }
    // verifica se a password está definida //
    if (trim(isset($_POST['userPassLogin']))){
        echo 0;
        return;
    }
    $parametros = [
        'nome_utilizador' => $_POST['nomeUserLogin'],
    ];
    $dao = new DaoUser(); // cria um objeto DaoUser //
    $dados = $dao->userLogin($parametros);
    if (count($dados) != 1) {
        echo 0;
        return;
    }
    else {
        if (password_verify(trim($_POST['userPassLogin']), $dados[0]->pass)) {
            $cliente = [
                'id_user' => base64_encode(Functions::pearl() . $dados[0]->id_user),
                'nome' => $dados[0]->nome,
                'foto' => $dados[0]->foto,
                'nome_utilizador' => $dados[0]->nome_utilizador,
                'pais' => $dados[0]->pais,
                'email' => $dados[0]->email,
            ];
            $dados = $_SESSION['cliente'] = $cliente;
            echo "Login efetuado!";
        }
    }
}
```

Figura 13 - Método login

A verificação do *hash* é efetuada através do método *password_verify(input, hash from SGBD)* onde recebe dois parâmetros, o input que vem do frontend e o *hash* do respetivo *user* que se encontra no SGBD. Se corresponderem é criada uma sessão com o nome *cliente* e com os dados do utilizador que são passados através do *array* associativo *\$cliente*.

4.8.Exemplo de prevenção de injeção de SQL

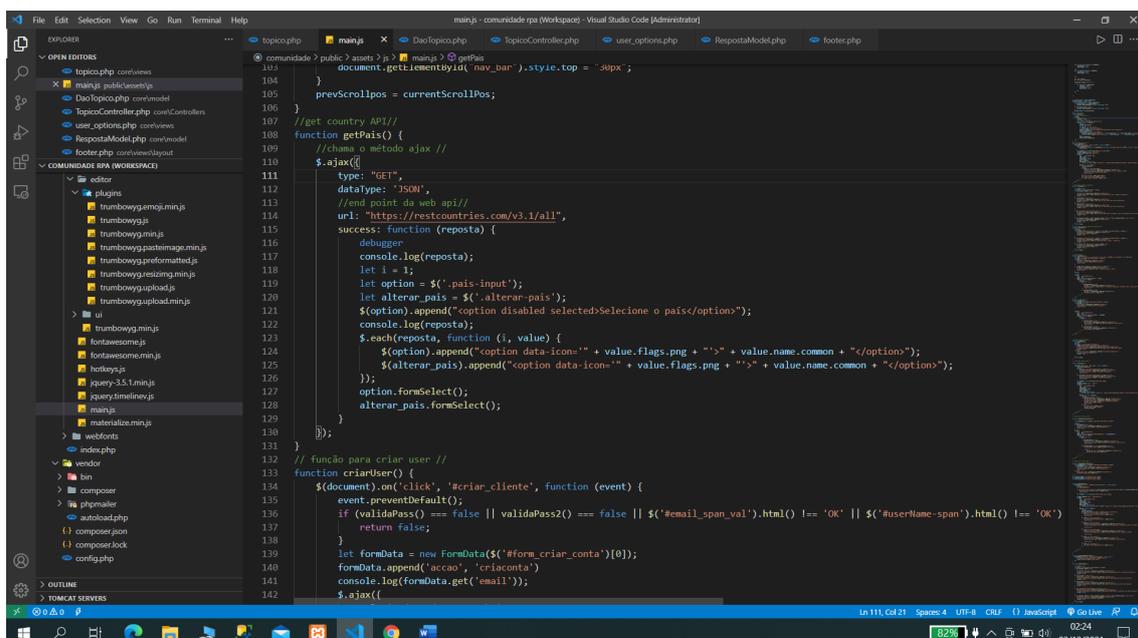


```
10 class DaoUser
11 {
12
13     public function verificaEmail($parametros)
14     {
15
16         //valida se já existe email na bd//
17         $sql = "SELECT email FROM users WHERE email = :email";
18
19         try {
20             $dados = [];
21             $ligacao = new Conexao();
22             $ligacao = $ligacao->ligacao();
23             $executar = $ligacao->ligacao();
24             $carrega = $executar->prepare($sql);
25             if ($parametros != null) {
26                 $carrega->execute($parametros);
27             } else {
28                 return;
29             }
30             if ($carrega->rowCount() > 0) {
31                 $dados = $carrega->fetchAll(PDO::FETCH_CLASS);
32             } else {
33                 $dados = [];
34             }
35         } catch (PDOException $e) {
36             $ligacao->desligar();
37             echo $e->getMessage();
38             return 0;
39         }
40         $ligacao->desligar();
41         return $dados;
42     }
43     public function verificaUser($parametros)
44     {
45         $validacao = 0;
46         $daoCrud = new DaoCrud();
47         //valida se já existe user na bd//
48         $retorno = $daoCrud->select("SELECT email FROM users WHERE nome_utilizador = :nome_utilizador", $parametros);
49         if (count($retorno) > 0) {
```

Figura 14 – Exemplo de prevenção de injeção de SQL

Na Figura 14, o método *verificaEmail()*, recebe como parâmetro um *array* associativo com os dados para executar a *query* de SQL. É efetuado uma consulta parametrizada com declarações preparadas conforme referenciado em 2.8.1, a fim de evitar a injeção de SQL.

4.9.Exemplo para consumir a API REST COUNTRIES



```
107 //get country API//
108 function getPais() {
109     //chama o método ajax //
110     $.ajax({
111         type: "GET",
112         dataType: "JSON",
113         //end point da web api//
114         url: "https://restcountries.com/v3.1/all",
115         success: function (reposta) {
116             debugger
117             console.log(reposta);
118             let i = 1;
119             let option = $(".pais-input");
120             let alterar_pais = $(".alterar-pais");
121             $(option).append("option disabled selected)Selecione o pais<option>");
122             console.log(reposta);
123             $.each(reposta, function (i, value) {
124                 $(option).append("<option data-icon='"+ value.flags.png + "'> + value.name.common + "</option>");
125             });
126             option.forSelect();
127             alterar_pais.forSelect();
128         }
129     });
130 }
131
132 // função para criar user //
133 function criarUser() {
134     $(document).on('click', '#criar_cliente', function (event) {
135         event.preventDefault();
136         if (validaPass() === false || validaPass2() === false || $('#email_span_val').html() != 'OK' || $('#userName-span').html() != 'OK')
137             return false;
138         let formData = new FormData($('#form_criar_conta')[0]);
139         formData.append('acao', 'criarconta');
140         console.log(formData.get('email'));
141         $.ajax({
```

Figura 15 – Exemplo para consumir API REST CUNTRIES via JavaScript

Para consumir a *API REST COUNTRIES* foi realizado, através de JavaScript, utilizando a biblioteca *Jquery*. Onde foi criada uma função com o nome *getPais()*, que invoca o método AJAX. Este recebe como parâmetros: o tipo de verbo HTTP que é o método *get*; o tipo de dados que são em JSON, e o *endpoint* *https://restcountries.com/v3.1/all*. Conforme a documentação da API, é devolvido um JSON com todos os países disponibilizados. De seguida os mesmos são apresentados na DOM.

4.10. Apresentação da aplicação web

4.10.1. Página inicial

A **Figura 16** representa a página inicial da *aplicação web – Comunidade RPA*

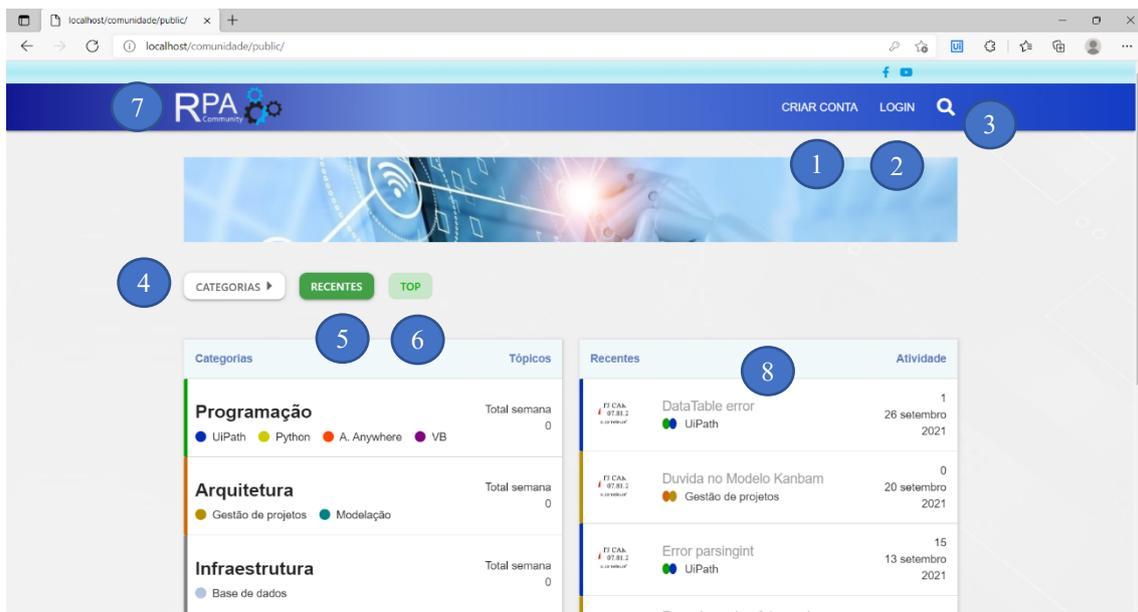


Figura 16 - Página inicial da aplicação (index.php)

Os pontos marcados na **Figura 16**, apresentam:

Ao clicar neste botão (1 – **Figura 16**) irá lançar um *modal* para criar uma conta de utilizador como é apresentado na **Figura 17**.

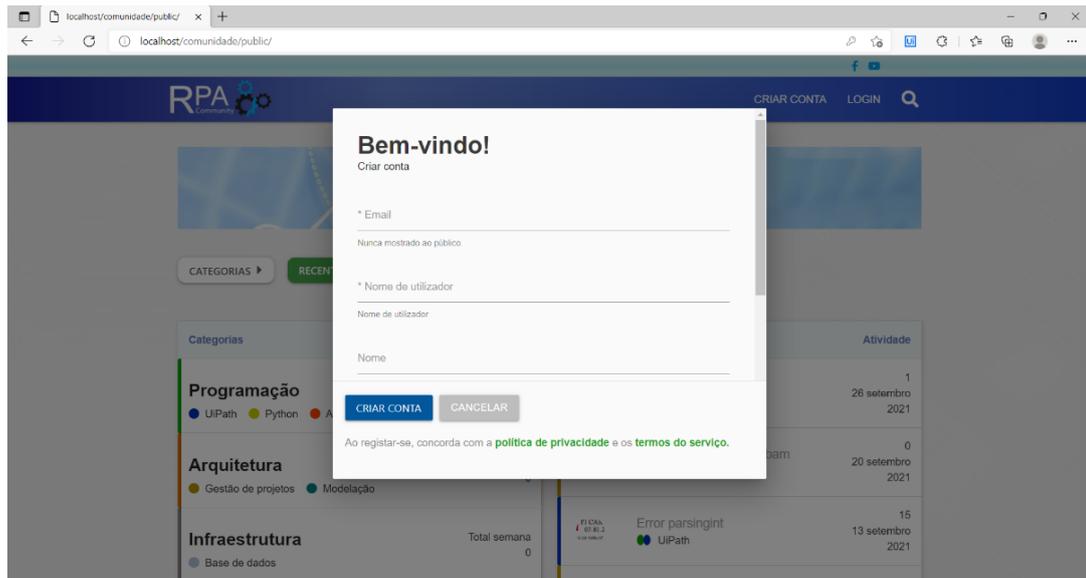


Figura 17 - Modal para criação de um utilizador

Após o registo, é enviado para o *email* do cliente um *purl* (*personal url*) com um link. A conta de utilizador só irá ficar ativa após clicar no *link* enviado (como apresenta a **Figura 18**).

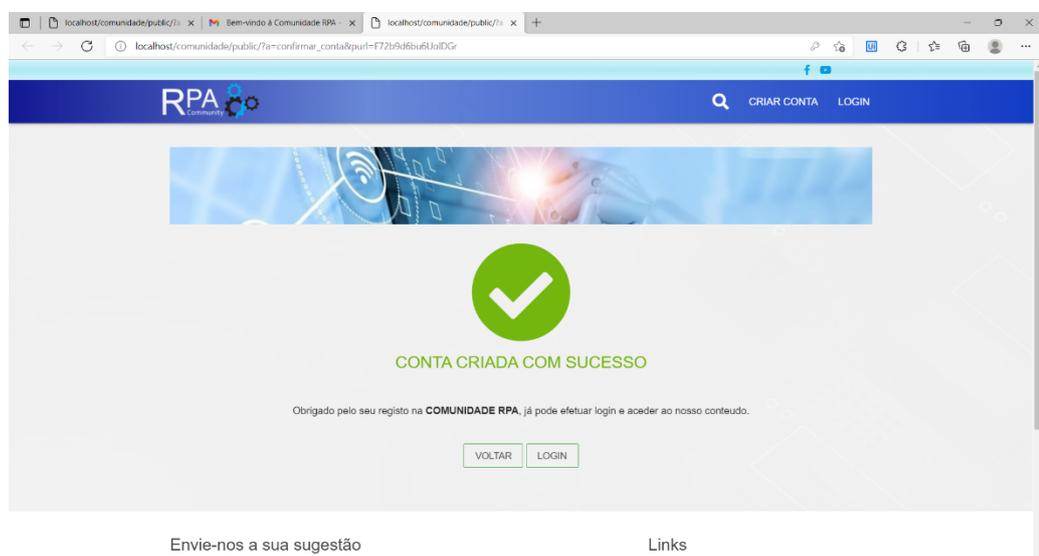


Figura 18 - Confirmação da conta

Ao clicar neste botão (2 – **Figura 16**) lançar-se-á um *modal* para um utilizador registado efetuar *login*, como apresenta a **Figura 19**.

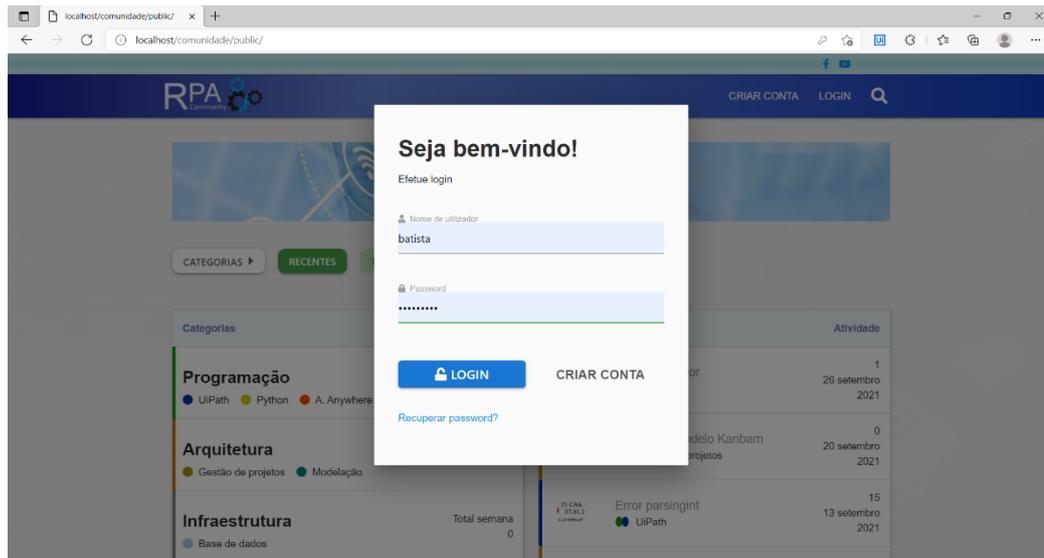


Figura 19 - Modal para login

Ao efetuar login o utilizador tem acesso à sua área de cliente, onde pode criar tópicos, *posts* e efetuar alterações de *crud* à sua conta, conforme mostra a Figura 20.

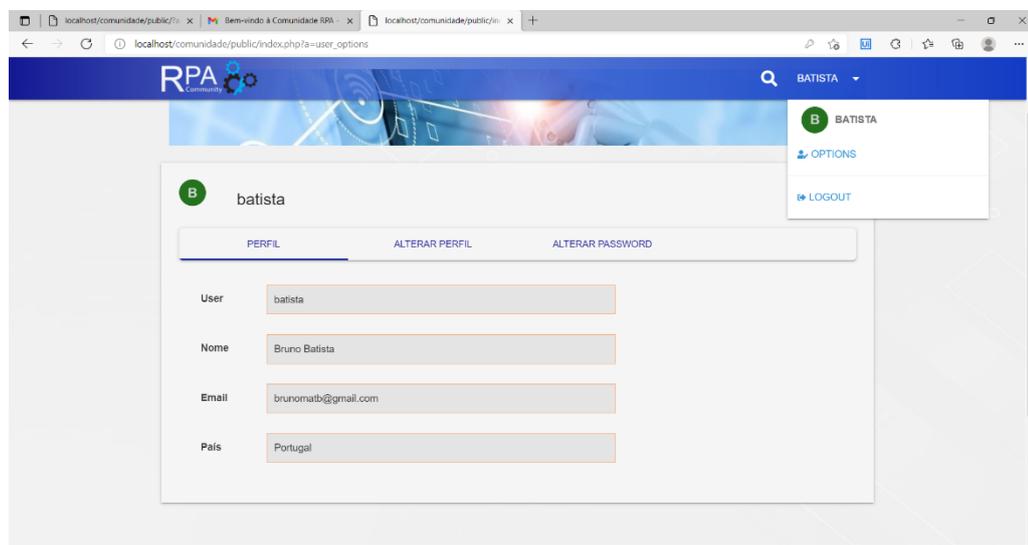


Figura 20 - Opções do utilizador

Ao clicar neste botão (3 – Figura 16) o utilizador vai poder efetuar consultas de tópicos por datas, como apresenta a Figura 21.

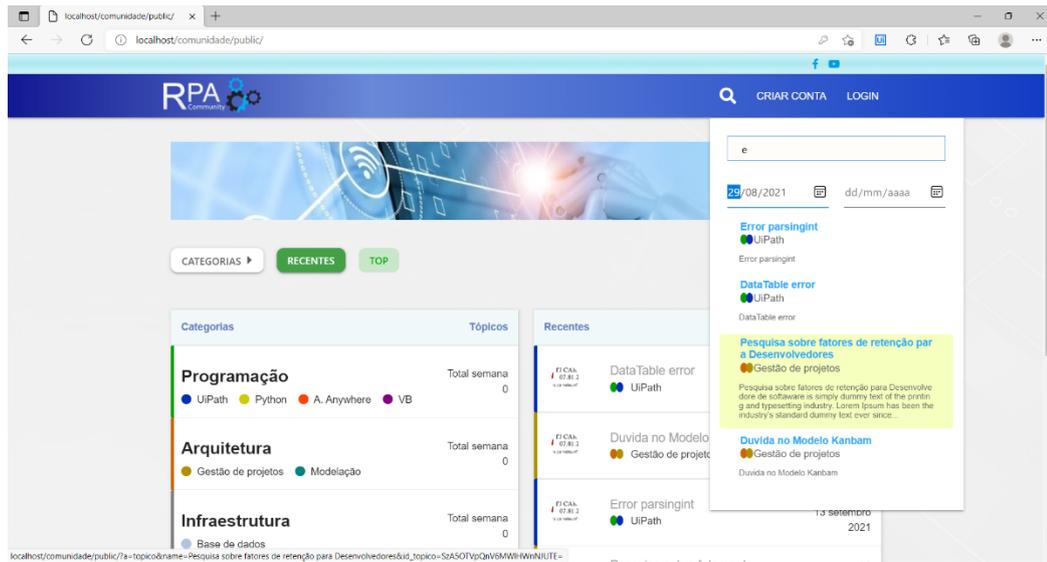


Figura 21 - Pesquisa de tópicos por datas

Ao clicar neste botão (4 – Figura 16) o utilizador vai ter acesso às categorias, e subcategorias, bem como pesquisa das mesmas, como apresenta a Figura 22.

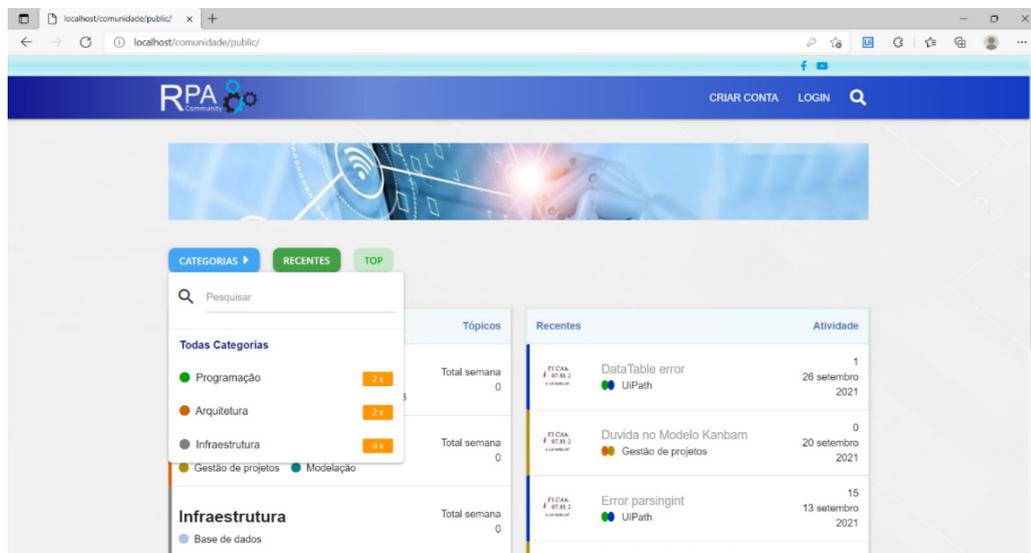


Figura 22 - Apresenta acesso às categorias e subcategorias

O utilizador pode navegar pelas categorias e subcategorias, e escolher um tópico conforme Figura 23.

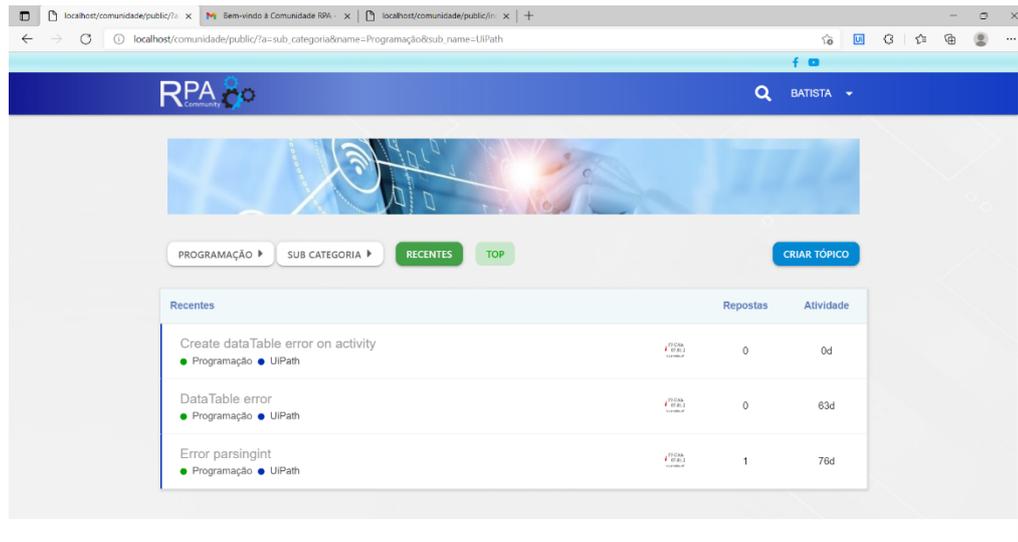


Figura 23 - Apresenta tópicos de uma determinada subcategoria

Ao clicar num tópico, caso seja o criador do mesmo pode editar, e efetuar *posts*, como apresenta a Figura 24.

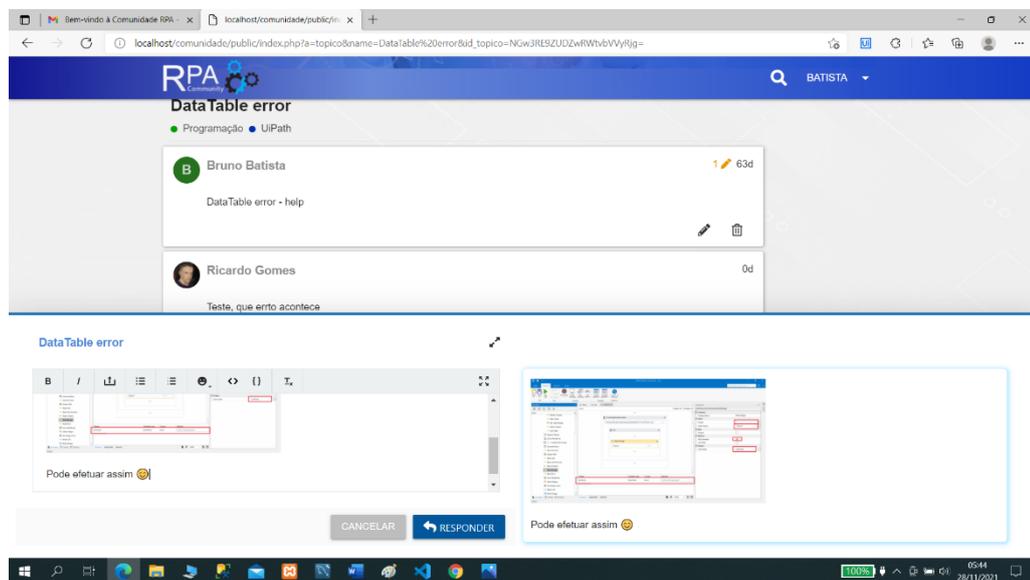


Figura 24 - Efetuar posts com preview

Ao clicar neste botão (5 – Figura 16) o utilizador é redirecionado para os tópicos mais recentes como apresenta a Figura 25.

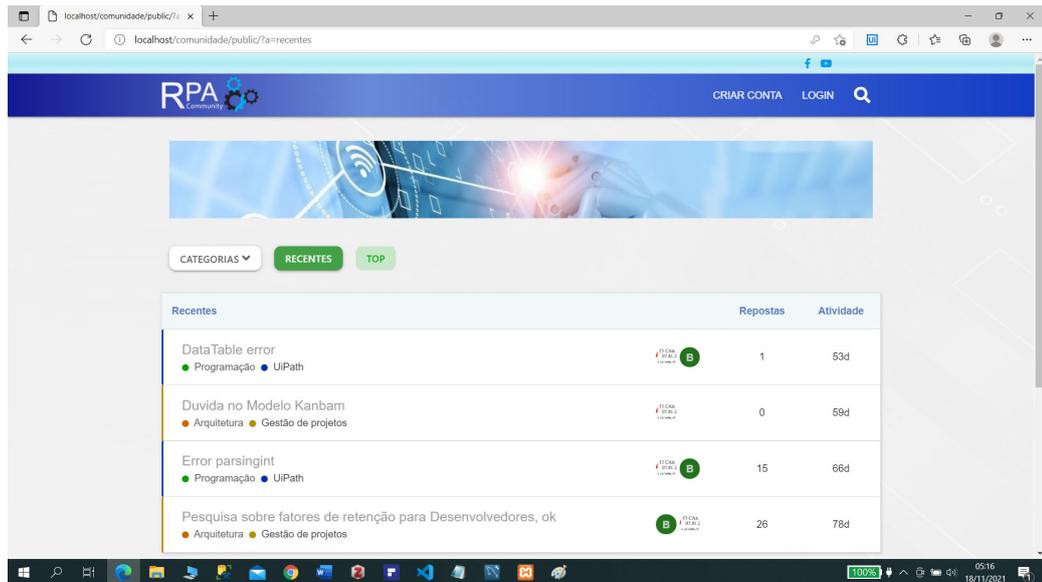


Figura 25 - Tópicos recentes

Ao clicar neste botão (6 – Figura 16) o utilizador é redirecionado para os tópicos com mais respostas (top) e ordenados por datas como apresenta a Figura 26.

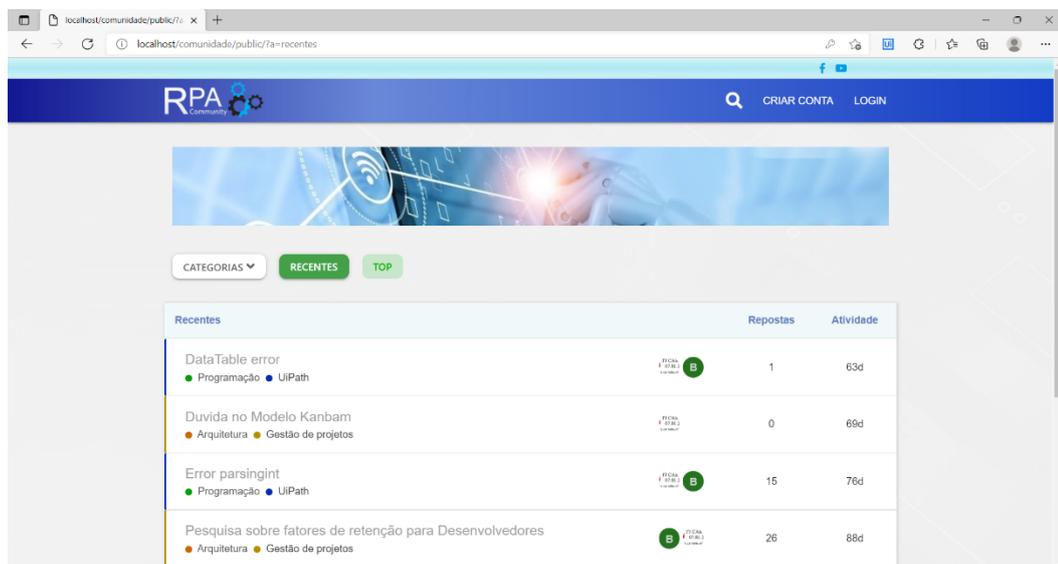


Figura 26 - Tópicos TOP

Logo da marca (7 – Figura 16) da aplicação web como apresenta a Figura 27.



Figura 27 - Logo da marca, criada em *Photoshop*.

5. Demonstração de resultados e análise

Nesta apresentar-se-ão os resultados efetuados através de um questionário e posteriormente a interpretação dos mesmos.

Género:
18 respostas

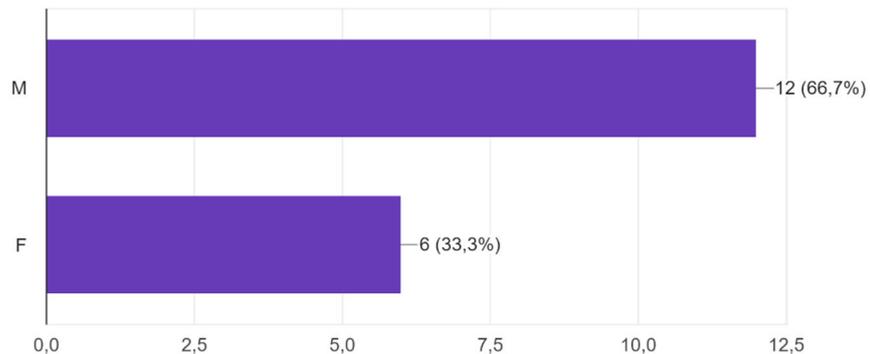


Figura 28 - Distribuição dos inquiridos

Verifica-se que 67,7% dos inquiridos são de sexo masculino e 33,3% de sexo feminino.

Idade:

18 respostas

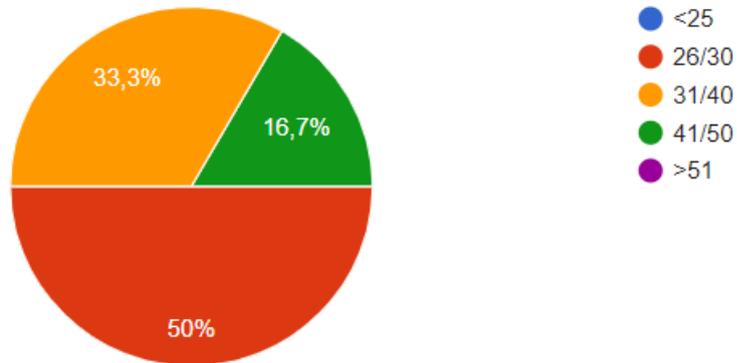


Figura 29 - Idade dos inquiridos

Verifica-se que 50% dos inquiridos têm idades compreendidas entre 26 e 30 anos de idade. Entre 31 e 40 anos de idade 33,3% e 16,0037% entre 41 e 50 anos de idade.

Habilitações:

18 respostas

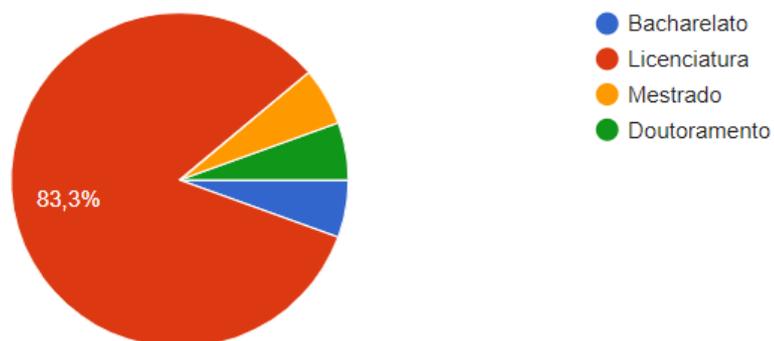


Figura 30 - Habilitações dos inquiridos

Em relação à terceira questão, 83,3% dos inquiridos possuem a licenciatura e apenas 5,6% distribuídos pelas seguintes habilitações: Bacharelato, Mestrado e doutoramento.

Considera que as comunidades virtuais são uma boa fonte de partilha de informação?

18 respostas

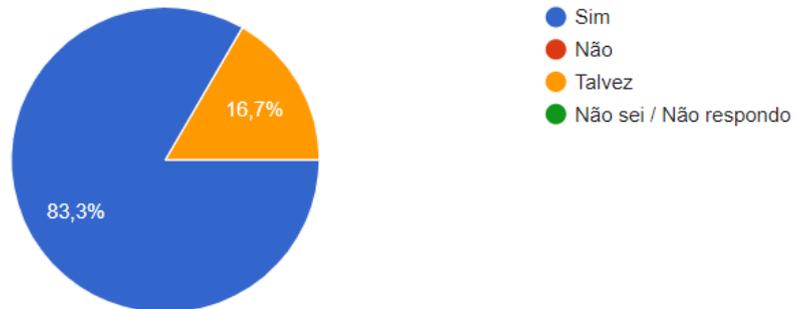


Figura 31 - Resposta à questão número 4

Em relação à quarta questão “*Considera que as comunidades virtuais são uma boa fonte de partilha de informação?*”, verifica-se que 83,3% dos inquiridos responderam de uma forma positiva e 16,7% “*Talvez*”.

Considera que as comunidades virtuais são uma boa fonte de aquisição de conhecimento?

18 respostas

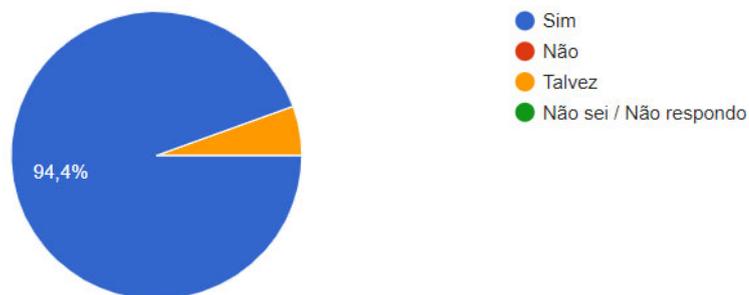


Figura 32 - Resposta à questão número 5

Em relação à quinta questão “*Considera que as comunidades virtuais são uma boa fonte de aquisição de conhecimento?*”, verifica-se que 94,4% dos inquiridos deram uma resposta positiva e 5,6% responderam “Talvez”.

Utiliza as comunidades virtuais para colocar questões?

17 respostas

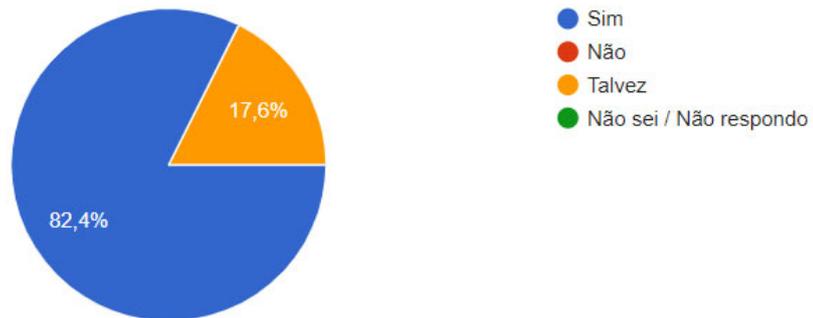


Figura 33 - Resposta à questão número 6

Em relação à sexta questão “*Utiliza as comunidades virtuais para colocar questões?*”, verifica-se que 82,4% dos inquiridos responderam positivamente e 17,6% deram uma resposta “Talvez”.

Utiliza as comunidades virtuais para partilhar informação com outros profissionais e / ou estudantes da área?

18 respostas

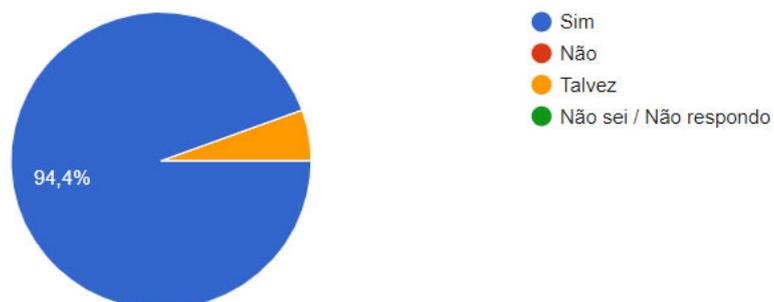


Figura 34- Resposta à questão número 7

Em relação à sétima questão “*Utiliza as comunidades virtuais para partilhar informação com outros profissionais e / ou estudantes da área?*”, verifica-se que 94,4% dos inquiridos têm uma resposta positiva e 5,6% têm uma resposta “Talvez”.

O conteúdo e partilha de informação já o ajudaram a nível profissional e / ou académico?

18 respostas

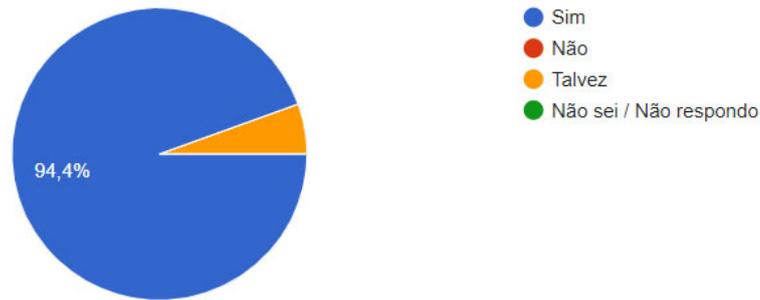


Figura 35 - Resposta à questão número 8

Referente à oitava questão “*O conteúdo e partilha de informação já o ajudaram a nível profissional e / ou académico?*”, verifica-se que 94,4% dos inquiridos deram uma resposta positiva e 5,6% indicaram que “Talvez”.

Nesta fase foi solicitado aos inquiridos para deslocarem-se ao www.comunidaderpa.pt e responderem as seguintes questões:

Considera que a aplicação web (comunidade RPA) é de fácil manuseamento?

18 respostas

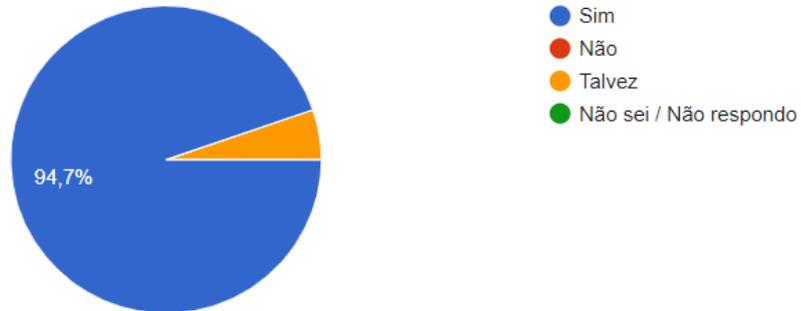


Figura 36 - Resposta à questão número 9

Referente à nona questão “*Considera que a aplicação web (comunidade RPA) é de fácil manuseamento?*”, verifica-se que 94,7% dos inquiridos responderam que “sim” e 5,3% deram uma resposta “Talvez”.

Considera que as informações disponíveis na aplicação web (comunidade RPA) são claras?

18 respostas

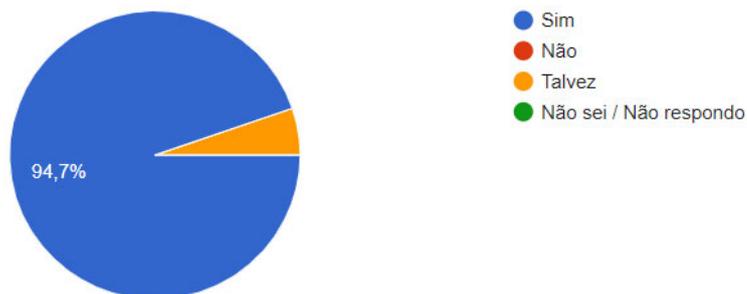


Figura 37 - Resposta à questão número 10

Referente à décima questão “*Considera que a aplicação web (comunidade RPA) é de fácil manuseamento?*”, verifica-se que 94,7% dos inquiridos têm uma resposta positiva e 5,3% têm uma resposta “Talvez”.

Em geral ficou satisfeito com a aplicação web (comunidade RPA) ?

18 respostas

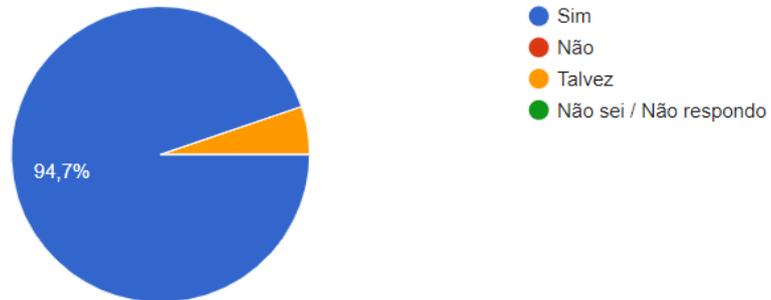


Figura 38 - Resposta à questão número 11

Referente à décima primeira questão “*Considera que as informações disponíveis na aplicação web (comunidade RPA) são claras?*”, verifica-se que 94,7% dos inquiridos têm uma resposta positiva e 5,3% têm uma resposta “Talvez”.

Numa escala de 0 a 5, qual a probabilidade de recomendar a aplicação web (comunidade RPA) ?

18 respostas

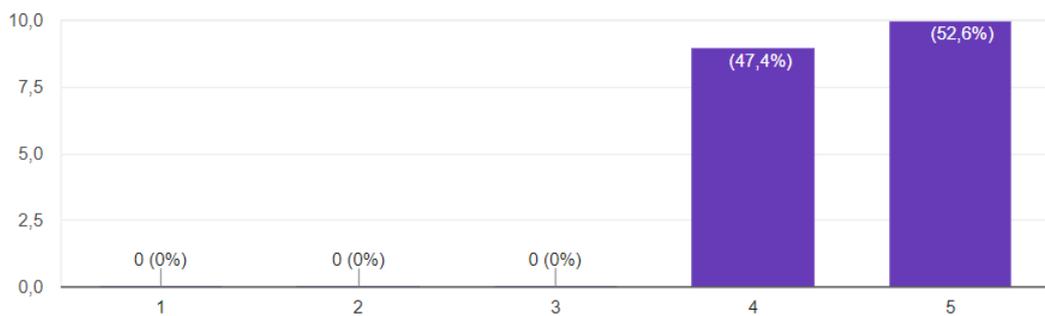


Figura 39 - Resposta à questão número 12

Referente à décima segunda e última questão “Numa escala de 0 a 5, qual a probabilidade de recomendar a aplicação web (comunidade RPA) ?”, 52,6% dos inquiridos responderam 5 e 47,4% responderam 4.

5.1. Interpretação dos resultados e conclusões

Após a verificação percentual dos resultados, seguir-se-á a análise e interpretação dos resultados, bem como as conclusões dos mesmos.

Este questionário foi efetuado a programadores profissionais de RPA, bem como estudantes da área.

Conforme a análise dos resultados obtidos, verifica-se que os inquiridos são maioritariamente do sexo masculino.

Referente ao gráfico da **Figura 29**, demonstra que a maioria dos inquiridos têm idades compreendidas entre 26 e 40 anos. Pelo que se trata de uma população maioritariamente jovem.

No gráfico da **Figura 30** verifica-se que os profissionais e estudantes de RPA que responderam ao questionário, a maioria possuem habilitações ao nível de licenciatura.

Analisando os gráficos da **Figura 31** e **Figura 32**, verifica-se que os inquiridos partilham e adquirem informação através das comunidades virtuais, sendo uma boa solução para interações entre profissionais e aprendizes de RPA.

Já a **Figura 33** e **Figura 34** vem consolidar a informação recolhida nos gráficos das figuras **Figura 31** e **Figura 32**, ou seja, os inquiridos fazem uso das comunidades virtuais para colocar questões, resolver problemas e partilhar informações. Assim sendo e como indica a **Figura 35** as comunidades virtuais ajudam no desenvolvimento, tanto profissional como académico.

De seguida efetuaram-se questões sobre o artefacto desenvolvido. Relativamente à usabilidade do site e conforme a **Figura 36**, conclui-se que os inquiridos não tiveram dificuldades na exploração e utilização da aplicação desenvolvida, sendo esta de fácil manuseamento conforme se aferiu.

Relativamente ao gráfico **Figura 37**, considera-se que a aplicação *web* dispõe de informação clara. Ou seja, no geral os inquiridos ficaram satisfeitos com o artefacto desenvolvido.

Conforme confirma a última questão, sendo uma das questões mais importante para a continuidade do projeto, os inquiridos não só ficaram satisfeitos como recomendariam a aplicação a outros profissionais da área.

Em suma as respostas obtidas foram positivas e de encontro às expectativas/objetivos, dilaniados inicialmente.

6. Reflexão final

O presente trabalho de final de curso possibilitou aprofundar e pôr em prática, conhecimentos e técnicas avançadas de desenvolvimento de aplicações *web*. Permitiu ainda aferir conhecimentos sobre a relevância do RPA nas empresas e nas suas respetivas áreas de negócio. Foi de extrema relevância refletir acerca da importância das comunidades virtuais, como partilha de informação e aquisição de saberes. Esta reflexão levou ao desenvolvimento de conhecimentos abrangentes, para poder encontrar o melhor caminho na realização da aplicação *web*.

Dando resposta ao objetivo inicial, conclui-se que as interações entre pessoas com interesses em comum e a contribuição da partilha de informação entre profissionais da área, são bastante essenciais. Estes aspetos possibilitam quebrar barreiras e otimizar o trabalho, bem como a aprendizagem.

Esta pesquisa teve ainda como objetivo aprofundar questões como: a segurança de aplicações *web*, arquitetura de *software*, os requisitos de engenharia do *software* (requisitos funcionais e não funcionais) e a modelação das bases de dados. Aferiu-se que todos estes aspetos são bastante importantes na realização das aplicações *web* e que as comunidades virtuais vieram tornar possível a partilha da informação, de uma forma mais célere e potencialmente eficaz.

Após a apresentação da aplicação que foi desenvolvida *Comunidade RPA* e de todas as funcionalidades que esta permite efetuar, verificou-se que foram de encontro às questões que se apresentaram na revisão bibliográfica.

Conclui-se que a maioria dos inquiridos considera este tipo de comunidade é essencial e facilitadora da partilha da informação e aquisição de conhecimentos. Foi ainda possível verificar que a apresentação da aplicação vai de encontro às expectativas dos utilizadores.

É de referir ainda que este tipo de trabalho exige muito tempo, dedicação, pesquisa, raciocínio lógico de todas as temáticas inerentes aos assuntos abordados e às linguagens de programação utilizadas.

Estes tipos de trabalhos são extremamente importantes pois, quando se encontra um problema e se definem objetivos, criam-se aptidões para encontrar as soluções mais viáveis e eficientes.

A realização deste trabalho deverá ter continuidade através da implementação e criação de novas funcionalidades, bem como a colocação do sistema desenvolvido em produção.

Bibliografia

- About—Materialize.* (2021). <https://materializecss.com/about.html>
- Alejandro Matos / Rest Countries.* (2021). GitLab. <https://gitlab.com/amatos/rest-countries>
- Bibeault, B., Katz, Y., & De Rosa, A. (2015). *JQuery in Action, Third Edition*. Manning Publications.
- Darie, C., & Brinzarea, e B. (2009). *AJAX and PHP: Building Modern Web Applications, 2nd Edition*.
- Delaney, C. (2020). Adaptive vs Responsive Design. *Ideas*. <https://xd.adobe.com/ideas/process/ui-design/adaptive-design-vs-responsive-design/>
- Edmunds, B. (2016). *Securing PHP Apps*. Apress.
- Frain, B. (2012). *Responsive Web Design with HTML5 and CSS3*. Packt Publishing Limited.
- Galloway, J., Wilson, B., Allen, K. S., & Matson, D. (2014). *Professional ASP.NET MVC 5*.
- Hevner, A., R, A., March, S., T, S., Park, Park, J., Ram, & Sudha. (2004). Design Science in Information Systems Research. *Management Information Systems Quarterly*, 28, 75.
- Hoy, M. B. (2011). HTML5: A New Standard for the Web. *Medical Reference Services Quarterly*, 30(1), 50–55. <https://doi.org/10.1080/02763869.2011.540212>
- IBM. (2021, Janeiro 6). *What is Data Modeling?* <https://www.ibm.com/cloud/learn/data-modeling>

Khalili, A., Auer, S., & Hladky, D. (2012). The RDFa Content Editor—From WYSIWYG to WYSIWYM. *2012 IEEE 36th Annual Computer Software and Applications Conference*, 531–540. <https://doi.org/10.1109/COMPSAC.2012.72>

Medeiros, O. L. (2016). *Segurança em aplicações Web*. <http://ric.cps.sp.gov.br/handle/123456789/271>

MySQL. (2020). <https://dev.mysql.com/doc/workbench/en/wb-intro.html>

Newman, S. (2019). *Monolith to Microservices*. O’ Reilly Media, Inc, USA.

PHPMailer—Uma classe de transferência e criação de e-mail com recursos completos para PHP. (2021). [PHP]. PHPMailer. <https://github.com/PHPMailer/PHPMailer>
(Original work published 2011)

Rheingold, H. (2021). *Virtual community | Definition, Characteristics, Types, & Facts | Britannica*. <https://www.britannica.com/topic/virtual-community>

Social Network for Programmers and Developers. (sem data). Obtido 13 de Novembro de 2021, de <https://morioh.com>

Sommerville, I. (2015). *Software-Engineering*.

Taulli, T. (2020). *The Robotic Process Automation Handbook*. APress.

Welling, L., & Thomson, L. (2016). *PHP and MySQL Web Development*. Pearson educação U.S.

Anexo A

Questionário de opinião sobre a importância das comunidades virtuais e do artefacto desenvolvido.

O presente trabalho, foi realizado no âmbito da Monografia da Licenciatura de Gestão de Sistemas e Computação da Universidade Atlântica tendo como tema *Aplicação Web – Comunidade de Robotic Process Automataion*.

Este questionário é destinado a profissionais e estudantes na área das TI.

1. Género:

Marcar tudo o que for aplicável.

- M
 F

2. Idade:

Marcar apenas uma oval.

- <25
 26/30
 31/40
 41/50
 >51

3. Habilitações:

Marcar apenas uma oval.

- Bacharelato
 Licenciatura
 Mestrado
 Doutoramento

4. É profissional da área de RPA?

Marcar apenas uma oval.

- Sim
 Não

5. Considera que as comunidades virtuais são uma boa fonte de partilha de informação?

Marcar apenas uma oval.

- Sim
- Não
- Talvez
- Não sei / Não respondo

6. Considera que as comunidades virtuais são uma boa fonte de aquisição de conhecimento?

Marcar apenas uma oval.

- Sim
- Não
- Talvez
- Não sei / Não respondo

7. Utiliza as comunidades virtuais para colocar questões?

Marcar apenas uma oval.

- Sim
- Não
- Talvez
- Não sei / Não respondo

8. Utiliza as comunidades virtuais para partilhar informação com outros profissionais e / ou estudantes da área?

Marcar apenas uma oval.

- Sim
- Não
- Talvez
- Não sei / Não respondo

9. Conteúdo e partilha de informação já o ajudaram a nível profissional e / ou académico?

Marcar apenas uma oval.

- Sim
- Não
- Talvez
- Não sei / Não respondo

Aceda à aplicação web através da URL <http://comunidaderpa.pt>

10. Considera que a aplicação web (comunidade RPA) é de fácil manuseamento?

Marcar apenas uma oval.

- Sim
- Não
- Talvez
- Não sei / Não respondo

11. É fácil encontrar a informação que pretende?

Marcar apenas uma oval.

- Sim
- Não
- Talvez
- Não sei / Não respondo

12. Considera que as informações disponíveis na aplicação web (comunidade RPA) são claras?

Marcar apenas uma oval.

- Sim
- Não
- Talvez
- Não sei / Não respondo

13. Em geral ficou satisfeito com a aplicação web (comunidade RPA) ?

Marcar apenas uma oval.

- Sim
- Não
- Talvez
- Não sei / Não respondo

14. Numa escala de 0 a 5, qual a probabilidade de recomendar a aplicação web (comunidade RPA) ?

Não recomendaria Recomendaria

Marcar apenas uma oval.

1 2 3 4 5