



ANTONIO MENEGHETTI FACULDADE - AMF

SISTEMAS DE INFORMAÇÃO

RHUAN MATHEUS DRESCHER

**DESENVOLVIMENTO DE SISTEMA DE OUVIDORIA PARA A ANTONIO MENEGHETTI
FACULDADE**

RESTINGA SÊCA

2019

RHUAN MATHEUS DRESCHER

**DESENVOLVIMENTO DE SISTEMA DE OUVIDORIA PARA A ANTONIO MENEGHETTI
FACULDADE**

Trabalho de Conclusão do Curso de Sistemas de
Informação, Antonio Meneghetti Faculdade – AMF
Orientador: Prof. Ms. João Otávio Cadó de Matos

RESTINGA SÊCA

2019

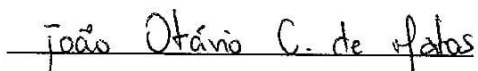
FACULDADE ANTONIO MENEGHETTI

Rhuan Matheus Drescher

DESENVOLVIMENTO DE SISTEMA DE OUVIDORIA PARA A ANTONIO
MENEGHETTI FACULDADE

Trabalho de Conclusão de Curso-Monografia, apresentado como requisito parcial para a obtenção do grau de Bacharel em Sistemas de Informação, Curso de Graduação em Sistemas de Informação, Faculdade Antonio Meneghetti-AMF.

Orientador: Prof. Ms. João Otávio Cadó de Matos



Prof. Ms. João Otávio Cadó de Matos

Orientador do Trabalho de Conclusão de Curso
Antonio Meneghetti Faculdade



Prof. Dr. Felipe Becker Nunes

Membro da Banca Examinadora

Antonio Meneghetti Faculdade



Prof. Ms. Samuel Vizzotto

Membro da Banca Examinadora

Antonio Meneghetti Faculdade

AGRADECIMENTOS

Quero agradecer a toda minha família, em especial aos meus pais, Dirceu Drescher e Naiara Drescher, por toda educação, incentivo e apoio dado nessa trajetória para que esse momento fosse possível.

A minha namorada, Gabrieli Silva, por estar ao meu lado neste importante momento, dando carinho e incentivo, sendo a maior apoiadora antes e durante este período.

Aos professores do curso de Sistemas de Informação, em especial ao meu orientador João Otávio, pelas preciosas orientações e conselhos, e ao professor Felipe Nunes, por proporcionar a oportunidade de iniciar um projeto desse nível em sala de aula e por acompanhá-lo durante todo esse tempo.

Ao meu grande amigo Sergio Lucas, que sempre me auxiliou tecnicamente.

Aos meus colegas e amigos, em especial ao André Menezes, Ezequiel Drews, Hercules Karkow, Jardel Knirsch, Letícia Machado e Luis Paulo Nery.

Por fim, à Antonio Meneghetti Faculdade e a todos que contribuíram e fizeram parte dessa importante etapa.

A todos, meu muito obrigado.

RESUMO

Instituições e empresas, de maneira geral, necessitam estar constantemente ouvindo a opinião de seus clientes para que seu negócio possa crescer e evoluir de acordo com sua visão. Dessa necessidade, surgem as ouvidorias, servindo como um canal de comunicação com o intuito de facilitar essa interação. Na Antonio Meneghetti Faculdade (AMF), utiliza-se um sistema de ouvidoria através de uma urna, um canal de difícil acesso e não muito seguro. Entende-se, então, que a instituição carece de um sistema mais moderno, com uma solução que contribua com o ambiente da Faculdade, criando um canal rápido, prático e seguro entre pessoas e a Instituição, aumentando a proximidade entre ambas as partes. Diante disso, o presente trabalho tem como objetivo o desenvolvimento de um sistema web de Ouvidoria, projetado para a AMF, onde foram utilizadas as linguagens de programação C# e TypeScript, frameworks e bibliotecas como ASP.NET Core e React e o padrão de desenvolvimento REST. Para tanto, foram elaborados diagramas UML, modelo entidade-relacionamento, dicionário de dados e foi desenvolvido um software seguindo o paradigma DDD e arquitetado em camadas, havendo assim a divisão de responsabilidades entre cada uma. O resultado do trabalho constitui-se em um sistema web, o qual foi apresentado suas telas e funcionalidades principais.

Palavras-chaves: Sistema de Ouvidoria, C#, TypeScript, React, ASP.NET Core, REST.

ABSTRACT

Institutions and companies, in general, need to be constantly listening the customers' opinions so that its business can grow up and evolve according to its vision. From this necessity the ombudsmen arise, serving as a communication chanel intending to facilitate this interaction. At Antonio Meneghetti Faculdade (AMF), an urn system is used to be its ombudsmen, a difficult to access and not very safe channel. Then it is understood that the institution lacks a more modern system, with a solution that contribute to the college environment, creating a fast, practical and safe channel between people and the institution, increasing the proximity between both parts. Considering this, this work has as objetctive the development of a ombudsmen web system, projected to the AMF, where the C # and TypeScript programming languages, frameworks and libraries such as ASP.NET Core and React and the REST development pattern were used. For this, UML diagrams, entity-relationship model and a data dictionary were elaborated and a software was developed following the DDD paradigm and architected in layers, having a responsibility division between each one. The final result of this work constitues in a web system that was presented its main pages and funcionalities.

Keywords: Ombudsmen System, C#, TypeScript, React, ASP.NET Core, REST.

LISTA DE FIGURAS

Figura 1 - Crescimento no uso de linguagens no GitHub.....	27
Figura 2 - Interface do Visual Studio Code	29
Figura 3 - Diagrama de casos de uso.....	32
Figura 4 - Tecnologias empregadas.....	33
Figura 5 - Camadas e projetos do sistema	34
Figura 6 - Modelo lógico	36
Figura 7 - Tela inicial	38
Figura 8 - Tela de login	39
Figura 9 - Tela de cadastro de usuário.....	40
Figura 10 - Tela de introdução às manifestações	41
Figura 11 - Tela de listagem de manifestações	42
Figura 12 - Tela de nova manifestação.....	43
Figura 13 - Tela de visualização de manifestação.....	44
Figura 14 - Tela de listagem de questionários	45
Figura 15 - Tela de responder questionário	46
Figura 16 - Tela administrativa de cursos.....	47
Figura 17 - Tela administrativa de usuários.....	48
Figura 18 - Tela administrativa de questionários	49
Figura 19 - Tela administrativa de cadastrar questionário.....	50
Figura 20 - Tela administrativa de relatório de questionários	51
Figura 21 - Tela administrativa de manifestações.....	52
Figura 22 - E-mail enviado pela AMF para divulgação do software e seus questionários	53
Figura 23 - Quadro de pontuação do SUS	54
Figura 24 - Respostas da pergunta 1.....	55
Figura 25 - Respostas da pergunta 2.....	56
Figura 26 - Respostas da pergunta 3.....	57
Figura 27 - Respostas da pergunta 4.....	58
Figura 28 - Respostas da pergunta 5.....	59
Figura 29 - Respostas da pergunta 6.....	60

LISTA DE GRÁFICOS

Gráfico 1 - Ranking das linguagens mais utilizadas no GitHub	24
Gráfico 2 - Busca no Google por bibliotecas JavaScript.....	26
Gráfico 3 - Ranking das <i>engines</i> de bancos de dados mais utilizados no mundo.....	28

LISTA DE ABREVIATURAS

AJAX – Asynchronous JavaScript And XML

AMF – Antonio Meneghetti Faculdade

API – *Application Programming Interface*

CSS – *Cascading Style Sheet*

DDD – *Domain Driven Design*

ES6 – ECMAScript 6

FAM – Fundação Antonio Meneghetti

HTML – *Hypertext Markup Language*

HTTP – *Hypertext Transfer Protocol*

HTTPS – *Hypertext Transfer Protocol Secure*

IoT – *Internet of Things*

JS – JavaScript

POO – Programação Orientada a Objetos

REST – *Representation State Transfer*

SQL – *Structured Query Language*

SUS – *System Usability Scale*

SPA – *Single Page Application*

TS – TypeScript

TSC – *TypeScript Compiler*

UML – *Unified Modeling Language*

URI – *Uniform Resource Identifier*

UX – *User Experience*

SUMÁRIO

1	INTRODUÇÃO.....	15
1.1	JUSTIFICATIVA.....	16
1.2	OBJETIVOS	17
1.2.1	Objetivo Principal	17
1.2.2	Objetivos Específicos	17
2	METODOLOGIA.....	18
3	ABORDAGEM TEÓRICA.....	19
3.1	Ouvidorias.....	19
3.1.1	Canais de comunicação	19
3.1.2	Funções de uma ouvidoria.....	20
3.1.3	Benefícios do uso	20
3.2	Tecnologias empregadas no desenvolvimento do projeto	21
3.2.1	.NET	21
3.2.2	.NET Core.....	21
3.2.3	ASP.NET Core.....	22
3.2.4	Programação Orientada a Objetos (POO)	22
3.2.5	C#	23
3.2.6	JavaScript (JS).....	23
3.2.7	Single Page Application (SPA).....	25
3.2.8	React	25
3.2.9	TypeScript.....	26
3.2.10	SQL Server	27
3.3	Ferramentas Utilizadas	28
3.3.1	Visual Studio Code.....	28

3.3.2	Azure Data Studio	29
4	O PROJETO DO SISTEMA DE OUVIDORIA	30
4.1	Requisitos funcionais e não funcionais	30
4.2	Elaboração de diagramas estruturais e comportamentais UML, modelo entidade relacionamento e dicionário de dados	30
4.3	Arquitetura e implementação	33
5	RESULTADOS	37
5.1	Tela Inicial	37
5.2	Login.....	38
5.3	Cadastro de Usuário	39
5.4	Introdução às Manifestações.....	40
5.5	Lista de Manifestações	41
5.6	Nova Manifestação.....	42
5.7	Visualização de Manifestação.....	43
5.8	Lista de Questionários	44
5.9	Responder Questionário	45
5.10	Telas Administrativas	46
5.10.1	Cursos e Departamentos	46
5.10.2	Usuários	47
5.10.3	Questionários	48
5.10.4	Cadastrar Questionário	49
5.10.5	Relatório de Questionários	50
5.10.6	Manifestações	51
6	DISCUSSÃO DOS RESULTADOS	53
6.1	Questionário de escala de usabilidade do sistema.....	54
6.2	Questionário de viabilidade do software	55

7	CONSIDERAÇÕES FINAIS	62
8	REFERÊNCIAS	63
	APÊNCIDE A	66
	APÊNDICE B	69

1 INTRODUÇÃO

Instituições, empresas e pessoas possuem papéis complementares que necessitam interagir dentro de uma sociedade. Essa interação permite que a sociedade evolua, promovendo, gradativamente, um melhor convívio entre seus agentes. Caso ocorram conflitos de opiniões, necessidades ou irregularidades, estas devem ser analisadas e uma alternativa deve ser adotada para recuperar a ordem e sanar os problemas, a fim de que o ambiente volte a ser agradável a todos ou à maioria. No entanto, essas manifestações precisam de um meio por onde possam ser relatadas e recebidas pelas pessoas certas – aquelas que têm poder de corrigir a situação. Para tanto, existem os sistemas de ouvidoria, servindo como um canal de comunicação para receber as manifestações dos usuários, categorizá-las e disponibilizá-las à instituição responsável.

Uma ouvidoria, portanto, corresponde a um meio disponibilizado por uma organização para ouvir o que os clientes têm a dizer – sejam denúncias, reclamações, solicitações, sugestões ou elogios – sobre os serviços que estão consumindo. Em outras palavras, trata-se de uma forma de receber o *feedback* dos clientes e posteriormente agir sobre eles. Para Rampinelli (2015), a ouvidoria, além de tudo, ajuda a organização a validar o quanto seus valores, diretrizes, padrões e missão estão sendo descumpridos e, então, apurar os fatos, descobrir os pontos de falha, realinhar estratégias e criar soluções. Cada instituição adota seu próprio modo de ouvir os clientes: telefone, internet e urnas são alguns dos mais comuns. A escolha do sistema para implantar a ouvidoria deve considerar o que mais seja acessível à realidade de seu público alvo.

Na Antonio Meneghetti Faculdade (AMF), Instituição de Ensino Superior localizada no município brasileiro de Restinga Sêca, o sistema de ouvidoria é disponibilizado através de uma urna posicionada no *hall* de entrada do prédio da Faculdade, onde é possível depositar uma carta para a Instituição fazendo a manifestação desejada. Porém, esse sistema não fica em um local muito visível, não é muito divulgado pela Instituição e não é de fácil acesso, pois caso alguém desejar relatar alguma manifestação, precisará se deslocar até a Faculdade para fazê-lo. Diante disso, é possível visualizar que a AMF carece de um sistema de ouvidoria mais atualizado, acessível e seguro, capaz de categorizar as manifestações para facilitar o tratamento, possibilitar às pessoas o envio de manifestações de qualquer lugar e garantir a segurança dos dados ali tratados, visto que um sistema por urnas pode ser facilmente invadido ou manipulado.

A partir desse cenário, compreende-se que uma plataforma online capaz de categorizar as manifestações, gerar relatórios gerenciais e que permita a aplicação de questionários conforme o interesse da Faculdade, com segurança, praticidade e privacidade, aumentaria a integração entre ouvidor e

comunidade acadêmica. Tendo esta necessidade em vista, o autor se propõe a aliar seus conhecimentos de programação web, adquiridos ao longo de sua trajetória acadêmica e profissional, ao desafio de suprir as carências do atual sistema de ouvidoria da AMF, vislumbrando facilitar a interação entre a instituição e seus frequentadores.

1.1 JUSTIFICATIVA

A motivação para o desenvolvimento desta solução surgiu inicialmente na disciplina de Programação Orientada a Objetos II, durante o semestre 2018/1, no curso de Sistemas de Informação da AMF, onde foi solicitado ao grupo de alunos, cujo o autor deste trabalho integrava, a criação de um protótipo de um sistema de ouvidoria com funcionalidades básicas, como a de realizar manifestações. Após o protótipo ser finalizado e aprovado em sua avaliação, o mesmo foi apresentado para representantes da Fundação Antonio Meneghetti (FAM) e da Antonio Meneghetti Faculdade, Sra. Any Rothmann, diretora comercial da FAM, Patrícia Wazlawick, responsável pelo setor Administrativo-Financeiro da AMF e Douglas Garlet, Gestor Comercial e responsável pela ouvidoria da Faculdade, que elogiaram o projeto desenvolvido e deram sugestões de funcionalidades adicionais que poderiam ser implementadas, como a aplicação de questionários e a extração de relatórios.

Após haver a aprovação do protótipo e o incentivo dos representantes das instituições relatadas anteriormente, o autor decidiu, mesmo após a conclusão da disciplina, dar continuidade ao projeto, vendo que a ideia poderia ser implementada. Porém, o que havia sido desenvolvido até então possuía muitos pontos de melhoria na visão técnica, como arquitetura, tecnologias e *user experience* (UX), que, na forma que estavam sendo utilizadas, poderiam dificultar a manutenção do sistema.

Diante disso, o autor resolveu iniciar um novo projeto, utilizando tecnologias multiplataforma, melhores práticas de programação e uma arquitetura em camadas, facilitando o desenvolvimento, manutenção e compreensão do código. Em seguida, procurou por possíveis falhas e pontos de melhoria em que o sistema de urna da AMF possuía, a fim de corrigi-los e implementá-los em seu novo software. Dentre eles, foram identificados problemas de segurança, podendo haver o uso de falsa identidade, de acessibilidade, visto que todas manifestações que forem feitas devem ser depositadas presencialmente na urna disponibilizada pela faculdade, de comunicação, no qual as pessoas que relatam suas manifestações não podem interagir e acompanhar o desenvolver dos seus relatos, e a possibilidade haver uma maior facilidade no gerenciamento das manifestações, por parte da Faculdade.

Perante as motivações e falhas apresentadas acerca do sistema utilizado atualmente, entende-se que a solução proposta neste trabalho trará uma maior comunicação entre a AMF e seu público (alunos,

professores, funcionários e visitantes), possibilitando um melhor ambiente de convívio a todos e estimulando a exposição de opiniões em relação à faculdade. Conforme Vilanova (2004, *apud* GLÜER, 2005):

A ouvidoria nas universidades constitui-se numa via desburocratizada e ágil para a solução de problemas e correção de injustiças, além de um importante canal de comunicação entre a comunidade universitária e a administração, contribuindo para o aperfeiçoamento do exercício da cidadania e para um constante *feedback*, essencial ao aprimoramento institucional e ao aumento de sua credibilidade.

1.2 OBJETIVOS

A fim de resolver o problema exposto anteriormente, este trabalho apresenta os seguintes objetivos:

1.2.1 Objetivo Principal

O principal objetivo do trabalho consiste no desenvolvimento de um sistema web de ouvidoria para a AMF, capaz de resolver os problemas relatados anteriormente e aumentar a proximidade das pessoas com a Instituição, tornando-a mais acessível, segura e moderna. Para isso, almeja-se construir uma solução que possibilite a categorização das manifestações, a extração de relatórios, a aplicação de questionários conforme interesse da Instituição, o acompanhamento das manifestações por parte dos usuários e a garantia de segurança e integridade dos dados manipulados.

1.2.2 Objetivos Específicos

- Trabalhar o sistema de ouvidoria também como uma plataforma que possibilite a construção de soluções de *Business Intelligence* e *Analytics* dentro da AMF, sobretudo a partir da geração de insights por meio de técnicas de análise de sentimentos das manifestações cadastradas;
- Disponibilizar, além das tradicionais ferramentas de avaliação institucional, um instrumento avaliativo mais informal e aberto;
- Desenvolver um sistema utilizando o padrão de desenvolvimento REST (*Representation State Transfer*) com um *front-end* SPA (*Single Page Application*);

2 METODOLOGIA

Este trabalho fez uso da pesquisa do tipo qualitativa do tipo experimental que, segundo Godoy (1995), busca enumerar e medir os dados estudados, não empregando instrumental estatístico na análise dos dados e que parte de questões ou focos de interesses amplos. Essa pesquisa é de natureza aplicada, onde se busca resolver um problema específico e de interesse local, tem objetivo exploratório, a fim de conquistar uma maior familiaridade com o problema (EVEN3, 201?), e foi realizado sob procedimentos de pesquisa-ação que, conforme Thiollent (1947, p. 14), é uma pesquisa com base empírica que é concebida e realizada em estreita associação com uma ação ou com a resolução de um problema coletivo e no qual os pesquisadores e participantes representativos da situação ou do problema estão envolvidos de modo cooperativo ou participativo.

Para alcançar os objetivos do projeto, foi feito o levantamento dos requisitos funcionais e não funcionais do sistema, criados diagramas de caso de uso, modelo entidade-relacionamento e dicionário de dados, foram escolhidas as linguagens de programação C# e TypeScript, frameworks como ASP.NET Core e React, SQL Server para o armazenamento de dados e também foi feita a escolha do padrão de desenvolvimento REST, seguindo o paradigma do *Domain Driven Design* (DDD) que, segundo Pires (2016), facilita a implementação de complexas regras e processos de negócios que são tratados como domínio. Para isso, o software foi arquitetado em camadas, sendo elas classificadas como de camadas de apresentação, aplicação, domínio, infraestrutura e transversais, visando facilitar a manutenção e separar suas responsabilidades.

Para avaliar os resultados desta pesquisa, foi realizada a aplicação de dois questionários que, segundo Gil (2008), é uma técnica de investigação composta por um conjunto de questões que são submetidas a pessoas com o propósito de obter informações. Neste trabalho, os questionários foram disponibilizados através do próprio software ao seu público alvo.

3 ABORDAGEM TEÓRICA

Neste item encontra-se toda a base teórica, que dá maior perspectiva a esse projeto. Está exposto um maior detalhamento sobre ouvidorias, as tecnologias que foram empregadas no desenvolvimento da aplicação, bem como seus conceitos, e as ferramentas utilizadas.

3.1 Ouvidorias

Uma ouvidoria é o setor de uma empresa destinado a ouvir seus clientes, com foco em solucionar problemas que não foram resolvidos por outros meios e coletar e tratar elogios e sugestões relatados por seus usuários. Ela tem o intuito de ser acionada somente quando os demais departamentos não conseguirem atender aquilo que está sendo tratado, servindo como uma ponte entre o público da empresa e ela própria, estando disponível para todas as pessoas acessarem e registrarem suas manifestações (RAMPINELLI, 2015).

Segundo CGU (2018, p. 12), as ouvidorias são instâncias de participação e controle social, responsáveis por interagir com os usuários, com o objetivo de aprimorar a gestão pública e melhorar os serviços oferecidos, garantindo os procedimentos de simplificação desses serviços.

Nos tópicos a seguir, serão abordados os canais de comunicação nos quais as ouvidorias podem utilizar, suas funções e também os benefícios que um sistema de ouvidoria pode trazer a uma instituição.

3.1.1 Canais de comunicação

Segundo CGU (2018, p. 28-30), para atingir seus objetivos, uma ouvidoria deve estimular o diálogo entre seus clientes e a empresa, podendo ocorrer através de diferentes canais de comunicação. Dentre eles, pode-se destacar meios como:

3.1.1.1 Meios eletrônicos

Canais que podem ser disponibilizados eletronicamente com o uso da internet. Para tanto, podem-se usar meios como e-mail, onde o cliente escreve um e-mail livremente e envia sua manifestação para um e-mail disponibilizado pela empresa, formulário eletrônico, no qual o cliente preenche um formulário com campos específicos disponibilizados em um endereço eletrônico e envia sua manifestação, redes sociais, onde o cliente relata sua manifestação através da rede social informada pela empresa, aplicativos, onde os relatos são enviados através de um aplicativo específico, bate papo, no qual

há uma janela de bate papo em algum site da empresa servindo como canal de relatos, ou então um chatbot, onde o cliente relata sua manifestação a um chat que utiliza inteligência artificial.

3.1.1.2 Meio orais

Uma ouvidoria pode também utilizar de meios orais, tratando via telefone e também pessoalmente, onde no primeiro o cliente utiliza um aparelho telefônico para ligar a um número disponibilizado pela empresa e, então, relata sua manifestação, e o segundo é realizado de maneira presencial ao ouvidor.

3.1.1.3 Meios físicos

Por fim, os meios físicos também podem ser utilizados como um canal de ouvidoria, podendo ser disponibilizado em cartas, onde há a livre escrita de uma manifestação e, então, o envio da mesma via serviços postais para um endereço fornecido pela empresa, depósito em urna ou entregue presencialmente na empresa, e também através de um formulário impresso, onde há o preenchimento de um formulário com campos pré-definidos pela empresa e o envio do mesmo à empresa.

3.1.2 Funções de uma ouvidoria

Para CGU (2018, p. 12-14), uma ouvidoria tem como suas funções ouvir e compreender as diferentes formas de manifestação dos cidadãos, reconhecer os usuários, sem qualquer distinção, como sujeitos de direitos, qualificar as expectativas dos usuários de forma adequada, caracterizando situações e identificando os seus contextos, para que o Estado possa utilizá-las como oportunidades de melhoria na gestão, dar tratamento adequado às demandas apresentadas pelos usuários, demonstrar os resultados produzidos em razão da participação social, produzindo informações capazes de subsidiar a gestão pública, acompanhar a prestação dos serviços públicos, com o objetivo de garantir sua efetividade, e promover a mediação e a conciliação.

3.1.3 Benefícios do uso

Com a disponibilização de um canal de ouvidoria, uma empresa aumenta seu poder de controle sobre situações boas ou indesejáveis, visualizando se os seus processos e serviços estão seguindo de acordo com seus valores, diretrizes, padrões e missões, a ponto de apurar os fatos, descobrir os pontos

de falha e sucesso, realinhar suas estratégias e criar soluções. Atuando como um complemento ao trabalho realizado pelos demais departamentos, ela age de forma independente e autônoma, lidando com questões delicadas.

Com ela, caso algo ocorra de maneira eficiente e satisfatória, a empresa pode analisar o caso de sucesso e tentar replicar em outras ocasiões e, caso contrário, pode analisar o caso de insucesso para tentar verificar os motivos que o geraram e então buscar uma solução (RAMPINELLI, 2015).

3.2 Tecnologias empregadas no desenvolvimento do projeto

3.2.1 .NET

O .NET é uma plataforma de desenvolvimento gratuita, multiplataforma e de código aberto desenvolvida pela Microsoft. Com ela, é possível escrever códigos em diferentes linguagens, como C#, F# e Visual Basic, abrangendo aplicações do tipo web, *mobile*, *desktop*, de microserviços, jogos, *machine learning*, nuvem e internet das coisas (IoT). Atualmente, a plataforma .NET possui três principais implementações, que são:

- .NET Core: implementação multiplataforma e de código aberto, possuindo suporte para Windows, Linux e macOS;
- .NET Framework: implementação que suporta somente desenvolvimento em Windows;
- Xamarin/Mono: implementação destinada a sistemas operacionais *mobile*;

Dentre essas três implementações, há também a .NET Standard, uma especificação formal de APIs que devem estar disponíveis em todas as implementações do .NET, a fim de estabelecer uma maior uniformidade entre todas elas (MICROSOFT, 2019).

3.2.2 .NET Core

O .NET Core é uma plataforma de desenvolvimento de código aberto, criada pela Microsoft. Ela possui suporte a multiplataforma, sendo compatível com Windows, macOS e Linux, e é mantida pela própria criadora e também pela comunidade .NET no GitHub, onde seu código fonte está versionado (MICROSOFT, 2018).

Seu desenvolvimento ocorreu devido ao fato do .NET Framework, a única implementação do .NET que não era focada em desenvolvimento *mobile*, estar disponível somente em sistemas Windows, assim, viu-se a necessidade de criar uma nova implementação do .NET, que suportasse as mesmas aplicações que o .NET Framework e que fosse multiplataforma.

3.2.3 ASP.NET Core

O ASP.NET Core é uma estrutura de código aberto disponível no GitHub, de multiplataforma e alto desempenho para criação de softwares modernos conectados à internet que pode ser utilizado para aplicativos e serviços web, IoT e *back-ends* móveis, tendo sua versão mais recente sendo a 3.0. Ele surgiu como uma evolução do ASP.NET 4.x, trazendo alterações de arquitetura que resultaram em uma estrutura mais enxuta e modular, com benefícios como a unificação de interfaces do usuário da web e das web APIs (*Application Programming Interface*), a capacidade de desenvolver e executar códigos em Windows, macOS e Linux e um pipeline de solicitação HTTP (*Hypertext Transfer Protocol*) leve, modular e de alto desempenho (MICROSOFT, 2019).

3.2.4 Programação Orientada a Objetos (POO)

A Programação Orientada a Objetos é um paradigma de programação que foi criado por Alan Kay e tem como objetivo aproximar o mundo real ao mundo virtual, buscando criar termos similares entre ambos, onde o principal termo encontrado foi “Objeto”, visto que todas as coisas que vemos pode ser definido como um objeto. Em sistemas que são desenvolvidos utilizando POO, os objetos (classes) que neles foram definidos e criados podem interagir entre si, onde cada um terá especificado o tipo de informação que pode enviar e receber (propriedades) e também as ações que ele mesmo pode realizar (métodos) (MONQUEIRO, 2007).

Gasparotto (2014) explica que na Programação Orientada a Objetos, existem quatro pilares principais, que são utilizadas como requisitos para uma linguagem ser considerada nesse paradigma, que são:

- Abstração: cada classe deverá ter uma identidade (ser única), possuir suas propriedades e métodos;
- Encapsulamento: define um nível de segurança de acesso às propriedades de uma classe, sendo elas privadas, protegidas ou públicas;
- Herança: consiste na possibilidade de uma classe herdar propriedades e métodos de outra classe e poder utilizá-las da mesma forma, caso elas não sejam privadas;
- Polimorfismo: consiste na possibilidade de uma classe poder alterar o comportamento ou funcionamento de um método herdado de outra classe.

Dessa forma, o desenvolvimento de um software se torna menos complexo, mais produtivo e compreensível, onde cada código criado possui uma responsabilidade e é feito de forma coesa com a realidade.

3.2.5 C#

O C# é uma linguagem de programação criada por Anders Hejlsberg, Scott Wiltamuth e Peter Golde, tendo sua primeira implementação amplamente distribuída lançada pela Microsoft em julho de 2000. Foi desenvolvida junto com a arquitetura .NET, tecnologia que também suporta linguagens como VB.NET, C++ e J#, e criada totalmente do zero para trabalhar na nova plataforma. Após o renome da linguagem, foi submetida pela Microsoft à ECMA (European Computer Manufacturers Association), associação que possui o objetivo de padronizar sistemas de informação. Foi aprovada em 2001, recebendo a especificação ECMA-334 e em 2003 também recebeu o padrão ISO, com a especificação ISO/IEC 23270 (ECMA, 2017).

Segundo Lotar (2010, p. 32), o C# foi evoluído das linguagens como C, C++ e Java e é uma linguagem de programação simples, orientada a objetos, poderosa e de fácil aprendizado, ideal para o desenvolvimento de aplicações web com ASP.NET.

O C# também possui suporte para programação orientada a componentes, pois o design de software atual depende cada vez mais dos componentes do software na forma de pacotes independentes e auto descritivos de funcionalidade, conforme Microsoft (2019). Desde seu lançamento, a linguagem recebeu vários *updates*, e atualmente encontra-se na versão 8.0 (MICROSOFT, 2017).

3.2.6 JavaScript (JS)

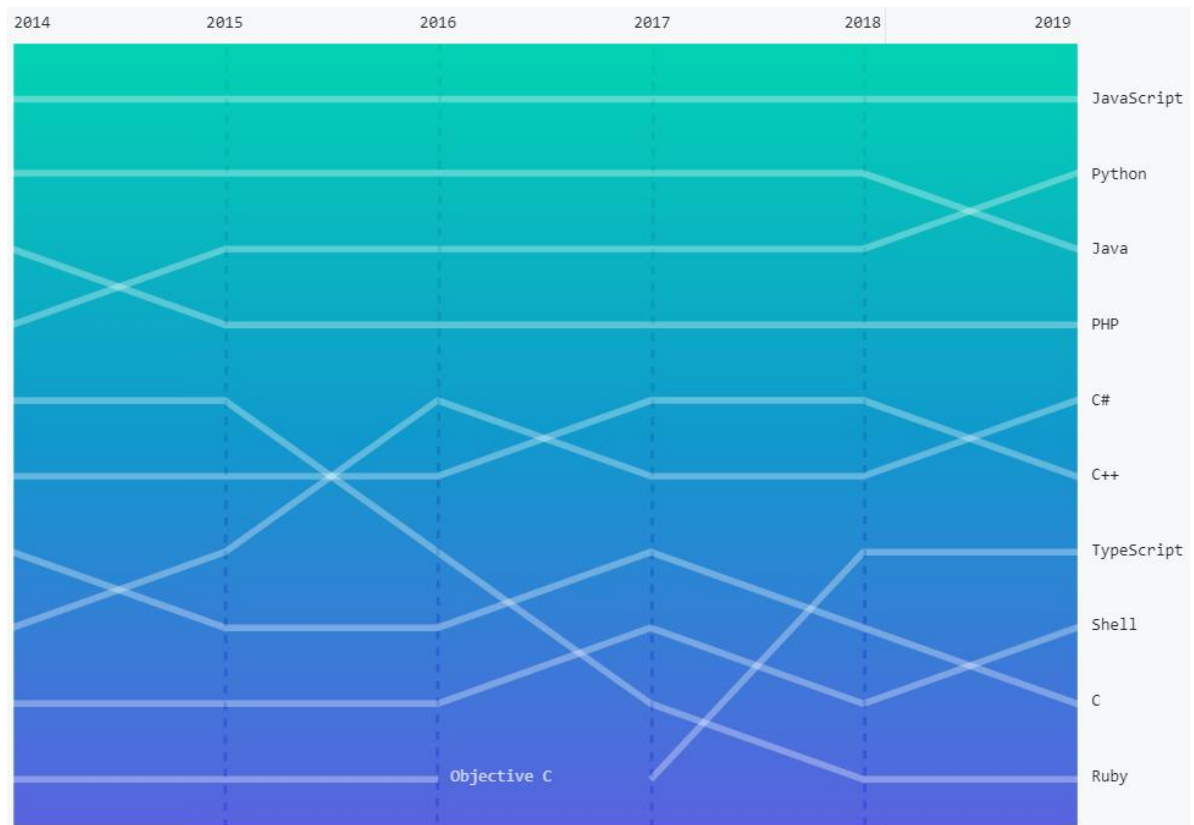
O JavaScript é uma linguagem de programação que, junto ao HTML (*Hypertext Markup Language*) e o CSS (*Cascading Style Sheet*), formam o conjunto de tecnologias que os navegadores entendem e conseguem interpretar. Com ela, é possível realizar ações que se atualizam dinamicamente nas páginas da web, como também controlar multimídias, imagens animadas e validar formulários (MDN, 2019).

Com seu crescimento, o JS deixou de ser uma linguagem que somente os navegadores conseguiam executar, para uma tecnologia que pode ser executada no desktop, como um programa comum, através do projeto Node.js, que permite não somente executar aplicações JavaScript no desktop,

como também trabalhar como servidor HTTP, possibilitando assim, trabalhar a linguagem tanto como cliente quanto servidor (MARCONDES, 2013).

Devido a esse crescimento e sua dinamicidade, o JavaScript se tornou umas das principais linguagens do mercado, estando na primeira posição das linguagens mais utilizadas no GitHub entre 2014 e 2019, conforme o Gráfico 1.

Gráfico 1 - Ranking das linguagens mais utilizadas no GitHub



Fonte: (GITHUB, 2019)

Com o crescimento do uso e da linguagem JavaScript, grandes empresas começaram a criar padrões, bibliotecas e técnicas de desenvolvimento para a linguagem, com o intuito de aumentar a performance de entrega de seus projetos. Como resultado, novos conceitos e nomenclaturas surgiram no ecossistema do JS, como SPAs, jQuery, React, AngularJS e Vue, trazendo as melhores práticas de desenvolvimento que essas empresas possuíam e disponibilizando para o uso da comunidade (MDN, 2019).

3.2.7 Single Page Application (SPA)

As *Single Page Applications* são aplicações desenvolvidas em JS que buscam levar toda a codificação do *front-end* para o cliente, onde o usuário navega entre as diferentes páginas do site sem realizar uma nova requisição ao servidor para que este retorne o código da página que está sendo acessada. Toda a codificação é carregada no primeiro acesso, onde o servidor envia os arquivos necessários ao navegador do cliente e ele fica encarregado de trabalhar com toda a parte de exibição, dando a impressão que se trata de uma aplicação desktop. A partir desse momento, as ações que terão necessidade de interação com o servidor serão tratadas através de requisições AJAX (Asynchronous JavaScript And XML).

Com o uso de SPAs, obtém-se uma melhor experiência de usuário, através da navegação sem requisições ao servidor, melhor desempenho, onde todo o sistema é carregado no primeiro acesso do cliente, e melhor manutenção, visto que o *front-end* agora é separado do *back-end* (LOCAWEB, 2016).

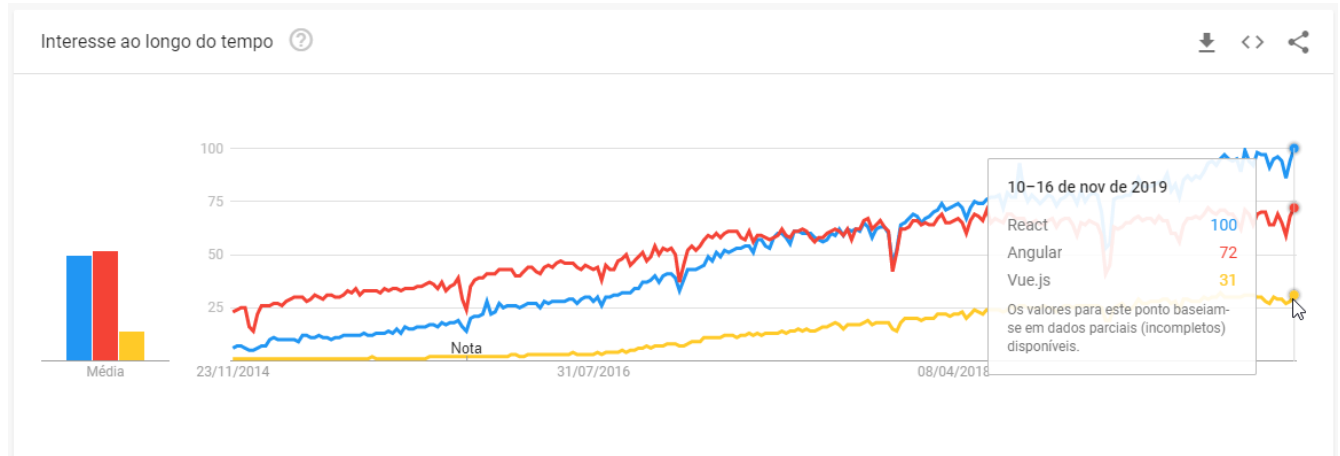
3.2.8 React

O React é uma biblioteca de JavaScript de código aberto, mantida pelo Facebook e Instagram e utilizada para criar aplicações SPA e que trabalha com componentes, onde cada componente possui um estado próprio, é renderizado de forma eficiente e somente quando necessário. Os componentes criados no React possibilitam que o desenvolvedor divida a interface do usuário em pequenos pedaços independentes, podendo ser reutilizados em outros componentes (REACT, 201?).

Essa biblioteca utiliza a sintaxe denominada JSX, uma extensão do JavaScript que consegue trabalhar e manipular elementos React dentro do código JS, o que facilita o desenvolvimento e o entendimento do código escrito, porém ela não é necessária para criar projetos (REACT, 201?).

Dentre as bibliotecas JavaScript mais conhecidas que utilizam SPA, o React vem se destacando mundialmente desde meados de 2017, onde suas buscas ultrapassaram a procura por Angular, conforme Gráfico 2.

Gráfico 2 - Busca no Google por bibliotecas JavaScript



Fonte: (TRENDS, 2019)

3.2.9 TypeScript

O TypeScript (TS) tem como principal desenvolvedor o mesmo criador da linguagem de programação C#, Anders Heijlsberg, junto a sua equipe da Microsoft. O projeto é *open source* e se baseia nos padrões do ES6 (ECMAScript), contando com a colaboração da comunidade técnica. Utilizando o TS, é possível fazer uso da programação orientada a objetos (POO), que tem como princípios o encapsulamento, herança, abstração e polimorfismo (DIONISIO, 2016).

Tratado como um *superset* – conjunto de funcionalidades exclusivas adicionais a outro conjunto base - de JavaScript, o TS é compilado para JS através do TypeScript Compiler (TSC), visto que os navegadores não suportam seus arquivos (LYCAN, 2017).

O TypeScript é uma das linguagens que mais tem crescido em questão de uso no GitHub, estando em quinto lugar no ranking divulgado em 2019, conforme demonstra a Figura 1.

Figura 1 - Crescimento no uso de linguagens no GitHub

01	Dart	532%
02	Rust	235%
03	HCL	213%
04	Kotlin	182%
05	TypeScript	161%
06	PowerShell	154%
07	Apex	154%
08	Python	151%
09	Assembly	149%
10	Go	147%

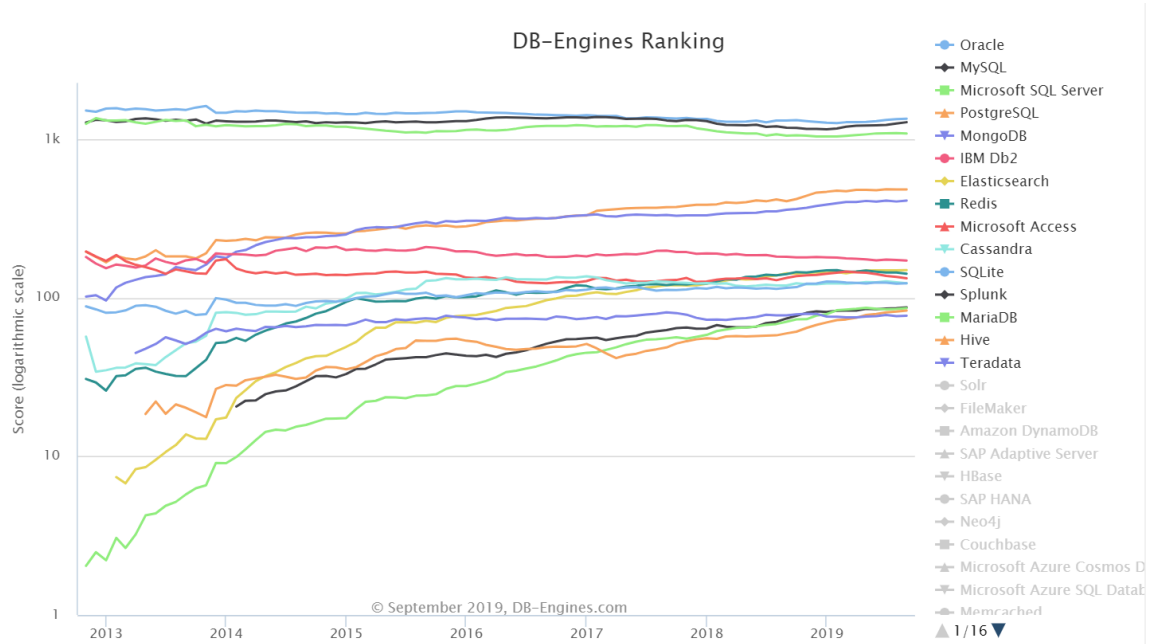
Fonte: (GITHUB, 2019)

Neste projeto, o TypeScript foi utilizado como linguagem de programação do projeto de apresentação, com o intuito de criar uma maior similaridade com o código C#, através das funcionalidades da Programação Orientada a Objetos presente em ambas linguagens, e também para facilitar futuras manutenções no código do cliente.

3.2.10 SQL Server

O SQL Server é um sistema de gerenciamento de banco de dados, fundada em 1988 pela Microsoft em parceria com a Sybase e que a partir de 1994, foi continuada somente pela Microsoft (SCHIMIGUEL, 2014). É um sistema de armazenamento relacional que possui versões pagas e gratuitas e é uma das mais utilizadas atualmente, conforme ranking apresentado no Gráfico 3.

Neste projeto, foi utilizada uma versão gratuita do SQL Server, para armazenar os dados que eram utilizados como teste no decorrer do desenvolvimento.

Gráfico 3 - Ranking das *engines* de bancos de dados mais utilizados no mundo

Fonte: (DB-ENGINES, 2019)

3.3 Ferramentas Utilizadas

3.3.1 Visual Studio Code

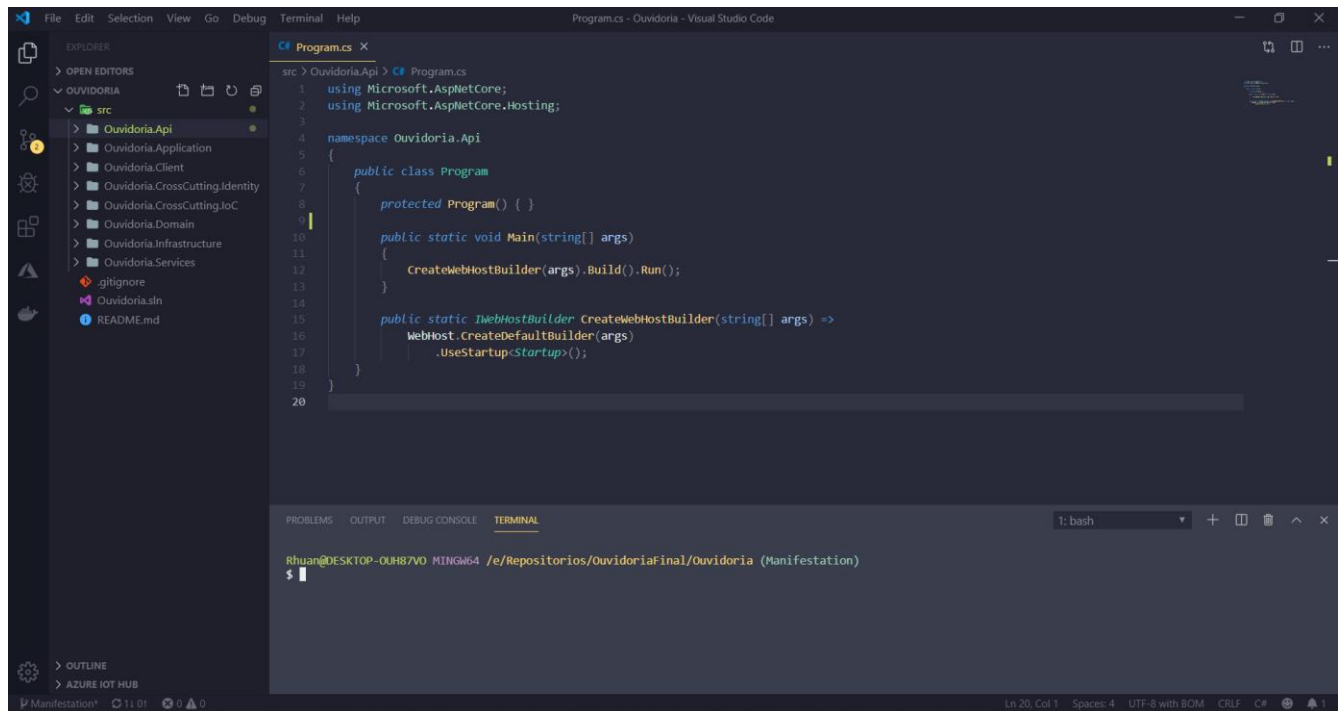
O Visual Studio Code é um editor de texto *open-source* desenvolvido pela Microsoft. É uma ferramenta multiplataforma para desenvolvimento de aplicações Web que dá várias possibilidades de desenvolvimento, pois possui um ecossistema de personalização rico que permite a instalação de extensões para desenvolvimento de diversas linguagens, para correções de sintaxe através do seu *intellisense* ou até mesmo para personalizar o próprio editor, utilizando temas desenvolvidos pela comunidade. Seu código fonte está “versionado” no GitHub (DIONISIO, 2016).

O Visual Studio Code foi utilizado durante todo o desenvolvimento do projeto, visto que possui extensões que suportam todas as linguagens de programação utilizadas, é facilmente personalizado de acordo com o desejo do usuário, possui um terminal integrado, o que facilitou a execução dos projetos do cliente e servidor de maneira simultânea, e também possui integração nativa com repositórios Git.

A Figura 2 exibe a interface do Visual Studio Code, onde se pode observar, ao centro da imagem, um código escrito em C# de um arquivo chamado “Program.cs”, um terminal aberto no local onde o projeto está localizado na máquina do usuário, na parte inferior da imagem, e uma barra de navegação

de opções do editor, na lateral direita da imagem, estando com a opção de “explorador de arquivos” selecionada.

Figura 2 - Interface do Visual Studio Code



Fonte: do autor (2019)

3.3.2 Azure Data Studio

O Azure Data Studio é uma ferramenta desenvolvida pela Microsoft para manipulação e manutenção da base de dados SQL. Assim como no Visual Studio Code, possui um bom sistema de intellisense, possibilidade de instalação de extensões e um terminal integrado. É uma ferramenta multiplataforma e *open-source*, cujo código fonte está “versionado” no GitHub (MICROSOFT, 2019).

4 O PROJETO DO SISTEMA DE OUVIDORIA

Para alcançar os objetivos do projeto, as seguintes etapas foram seguidas:

4.1 Requisitos funcionais e não funcionais

Inicialmente foi realizado o levantamento e a definição dos requisitos funcionais e não funcionais que deveriam estar presentes no desenvolvimento do projeto. Estes requisitos foram definidos na disciplina de Programação Orientada a Objetos II, durante o semestre 2018/1, no curso de Sistemas de Informação da AMF e aprimorados pelo autor após o encerramento da disciplina.

Dentre os requisitos funcionais que o sistema deve possuir, destacam-se:

- Cadastros de cursos, departamentos, usuários, manifestações e questionários;
- Alteração nos dados de cursos, departamentos e usuários;
- Exclusão de registros de cursos e departamentos;
- Ativação e inativação de acessos dos usuários;
- Listagens e consultas de dados cadastrados no sistema;
- Relatórios gerenciais;
- Interações entre manifestante e ouvidor nas manifestações cadastradas;
- Responder questionários.

Já dentre os requisitos não funcionais que o sistema deve atender, destacam-se:

- Sistema web compatível com diferentes navegadores;
- Diferentes níveis de permissões, sendo inicialmente em nível de usuário e administrador;
- Autenticação de usuário;
- Design responsivo;
- Arquitetura dividida em camadas;
- O projeto deverá poder ser executado em diferentes sistemas operacionais;
- O projeto deverá poder ser publicado em diferentes sistemas operacionais.

4.2 Elaboração de diagramas estruturais e comportamentais UML, modelo entidade relacionamento e dicionário de dados

Após a definição dos requisitos funcionais e não funcionais, iniciou-se a estruturação de diagramas da UML (*Unified Modeling Language*), uma linguagem gráfica para visualização, construção e documentação de softwares (BOOCH, RUMBAUGH e JACOBSON, 2005). Os diagramas definidos

foram o de classes e o de casos de uso que, segundo Dos Santos *et al* (2006), é um documento narrativo que descreve a sequência de eventos de um ator (um agente externo) que usa um sistema para completar um processo. Este último representado em sua forma resumida na Figura 3, onde pode-se observar as funcionalidades e perfis de usuário disponíveis no sistema, representados, respectivamente, pelos casos de uso e atores.

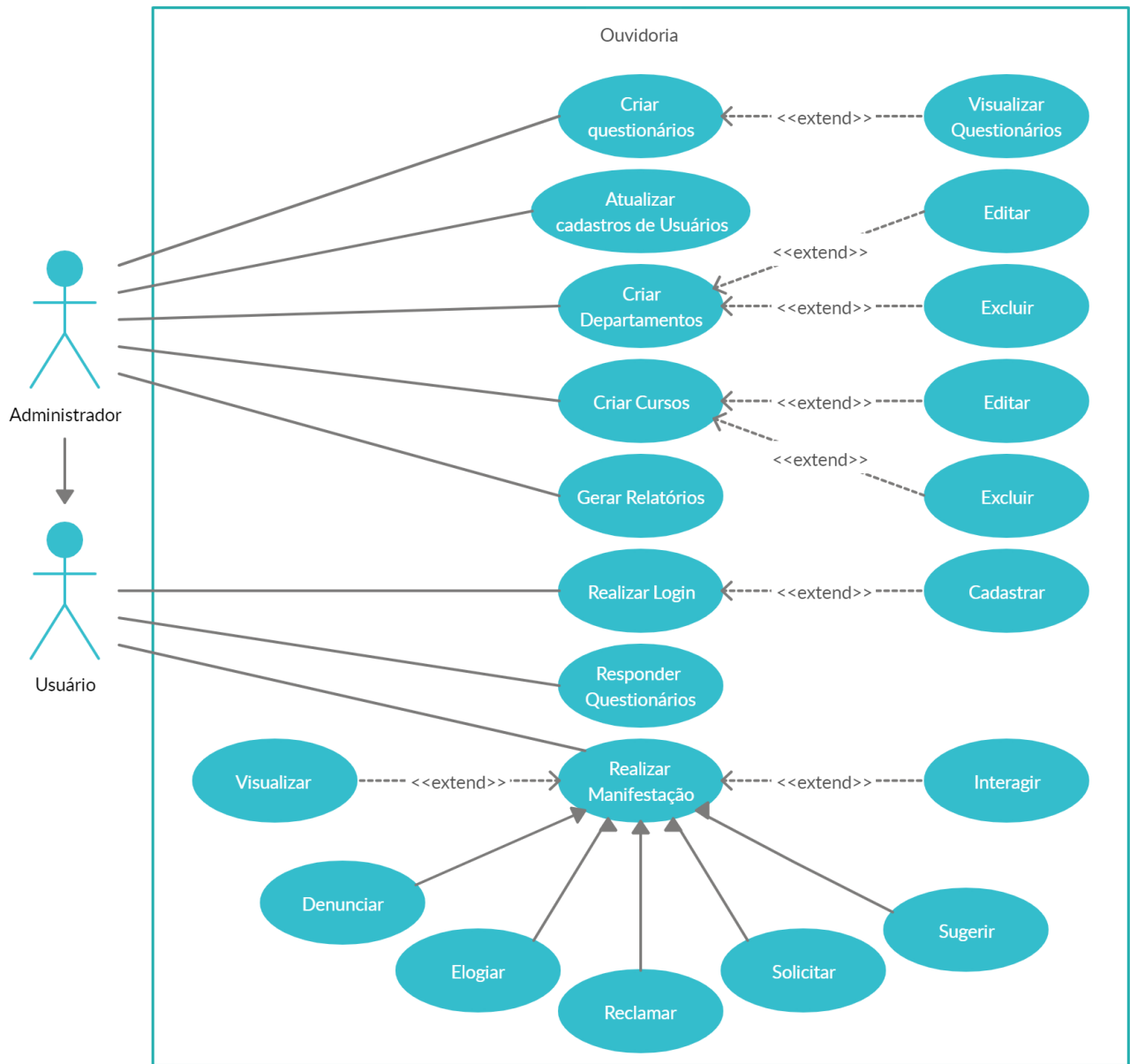
Para administradores, as seguintes ações funcionalidades são disponibilizadas:

- Cadastro, edição e exclusão de cursos e departamentos;
- Cadastro e visualização de questionários;
- Edição e inativação de cadastros dos usuários;
- Geração de relatórios;
- Todas as ações disponíveis para usuários.

Já para usuários, são disponibilizadas as seguintes funcionalidades:

- Cadastro de manifestações de cinco tipos: Denúncias, Elogios, Reclamações, Solicitações e Sugestões;
- Visualizar e interagir com suas manifestações, inserindo comentários e respostas a comentários já realizados a elas;
- Responder aos questionários;
- Realizar *login*;
- Cadastrar-se.

Figura 3 - Diagrama de casos de uso



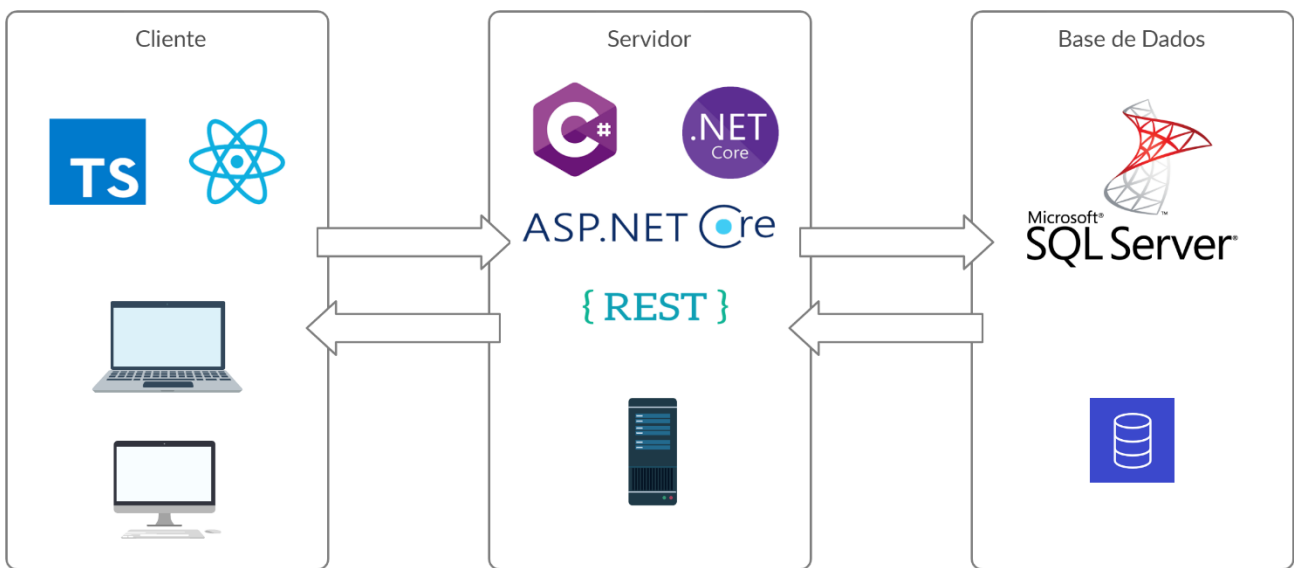
Fonte: do autor (2019)

Na sequência, estabeleceram-se o modelo entidade-relacionamento e seu dicionário de dados, contendo todas as tabelas, suas ligações e seus respectivos atributos, elaborando, assim, todas as entidades do domínio da aplicação que deveriam estar no desenvolvimento do software.

4.3 Arquitetura e implementação

Com os requisitos, diagramas e modelos finalizados, partiu-se para a etapa de definição da arquitetura do software e das tecnologias que seriam utilizadas no desenvolvimento do mesmo. Como padrão de desenvolvimento, adotou-se o REST, um conjunto de princípios e restrições que definem formas de uso do HTTP e URIS com o intuito de resolver problemas de integração entre aplicativos e simplificar estruturas arquiteturais orientadas a serviços (ALLET, 2013), construindo, assim, uma API que possa ser utilizada pela faculdade não somente para este sistema, mas também em qualquer outro que necessite consumir seus dados, seja ele web, *mobile* ou *desktop*. Como tecnologias, adotaram-se a linguagem de programação C# e o framework ASP.NET Core para o *back-end*, enquanto a biblioteca de JavaScript React e o *superset* TypeScript foram os escolhidos para o desenvolvimento do *front end* e o SQL Server como base de dados, conforme ilustrado na Figura 4.

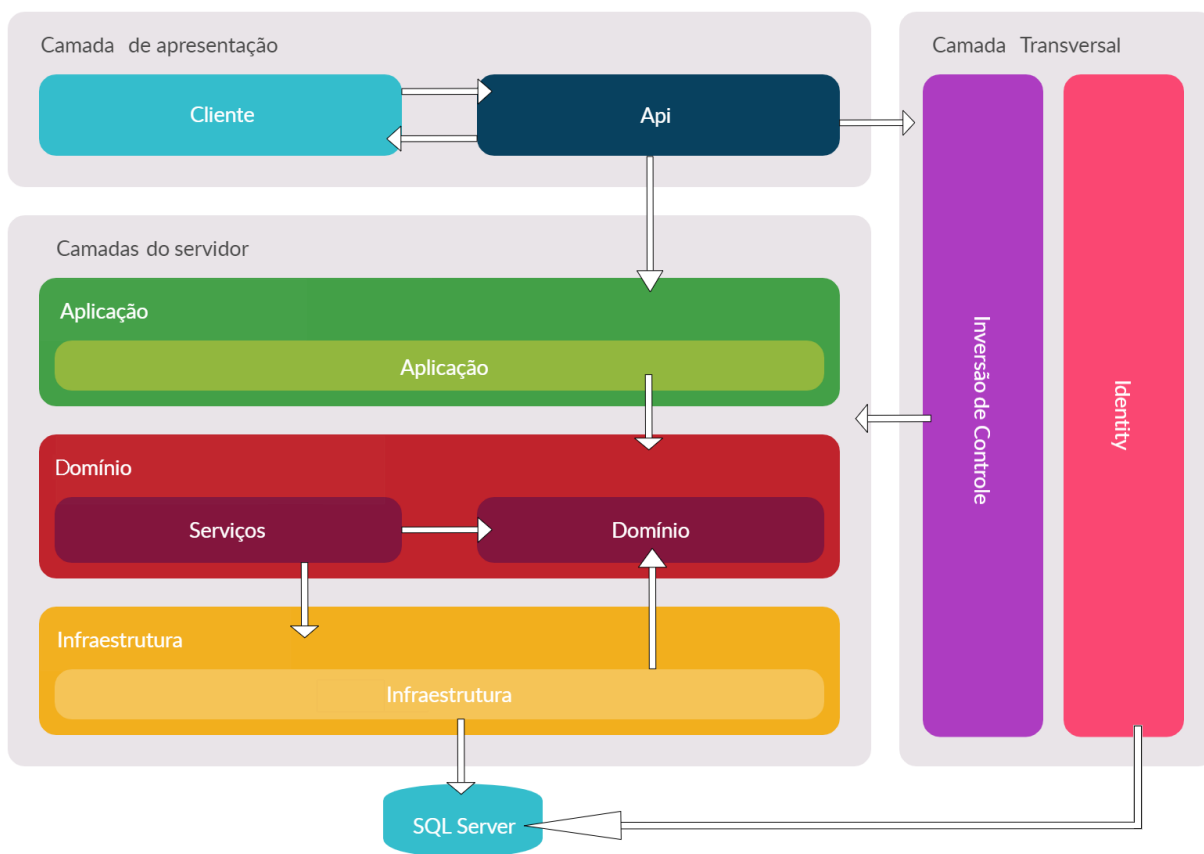
Figura 4 - Tecnologias empregadas



Fonte: do autor (2019)

A partir disso, iniciou-se o desenvolvimento do software onde, visando facilitar o desenvolvimento e a implementação de complexas regras e processos de negócio, adotou-se a abordagem de design denominada DDD, através da qual o desenvolvimento é guiado pelo domínio da aplicação, ou seja, focado no negócio. Para tanto, a solução criada foi arquitetada de forma que fosse separada em camadas, sendo estas de apresentação, aplicação, domínio, infraestrutura e transversal, contendo oito diferentes projetos, conforme Figura 5.

Figura 5 - Camadas e projetos do sistema



Fonte: do autor (2019)

Com essa separação em camadas, é possível dividir as responsabilidades do software, facilitando sua manutenção e aumentando sua escalabilidade, visto que cada uma delas está encarregada de realizar somente suas respectivas tarefas. Essa separação foi feita da seguinte forma:

1. Camada de apresentação:

Responsável por conter todos os projetos que o usuário consegue interagir. Nela, ficam dois projetos:

- **Cliente:**

Responsável por toda a parte visual que o usuário interage, nela estão contidos todo o código TypeScript, os componentes e páginas React e as rotas de navegação do sistema.

- **API:**

Projeto onde ficam todas as *Controllers*, com as rotas do servidor, e configurações da aplicação, isto é, a inicialização das bibliotecas e a definição

de políticas que a aplicação irá utilizar, como o uso de HTTPS (Hyper Text Transfer Protocol Secure) como protocolo de transferência de dados.

2. Camada de aplicação:

Faz a ligação entre as solicitações do usuário e as regras de negócio do sistema, possuindo um único projeto, cujo nome é o da própria camada. Nele, estão os modelos de apresentação do usuário (*ViewModels*), configurações de mapeamento entre estas entidades e as de domínio, as interfaces e implementações dos serviços de aplicação.

3. Camada de domínio:

É a camada principal de todo o software, pois é nela que estão contidas as regras de negócio do sistema e entidades e suas validações. Possui dois projetos:

- Domínio:

Projeto que possui as entidades, enumeradores, interfaces de repositórios e validações das entidades.

- Serviços:

É o projeto responsável por tratar as regras de negócio e realizar validações que necessitem de acesso ao banco de dados. Nele ficam todas as interfaces e implementações dos serviços de domínio.

4. Camada de infraestrutura:

Responsável por tratar unicamente de toda a conexão e interação com o banco de dados, possuindo apenas um projeto, nomeado com o mesmo nome da camada. Neste projeto, ficam todas as implementações de repositórios, contextos do domínio, as migrações do banco de dados e também todos os mapeamentos das entidades com a base.

5. Camada transversal:

É uma camada que serve de apoio às demais camadas, podendo ter visão de uma ou mais camadas de forma com que não influencie diretamente no domínio da solução. Possui dois projetos:

- Inversão de controle:

Responsável por fazer todas as injeções de dependência do projeto.

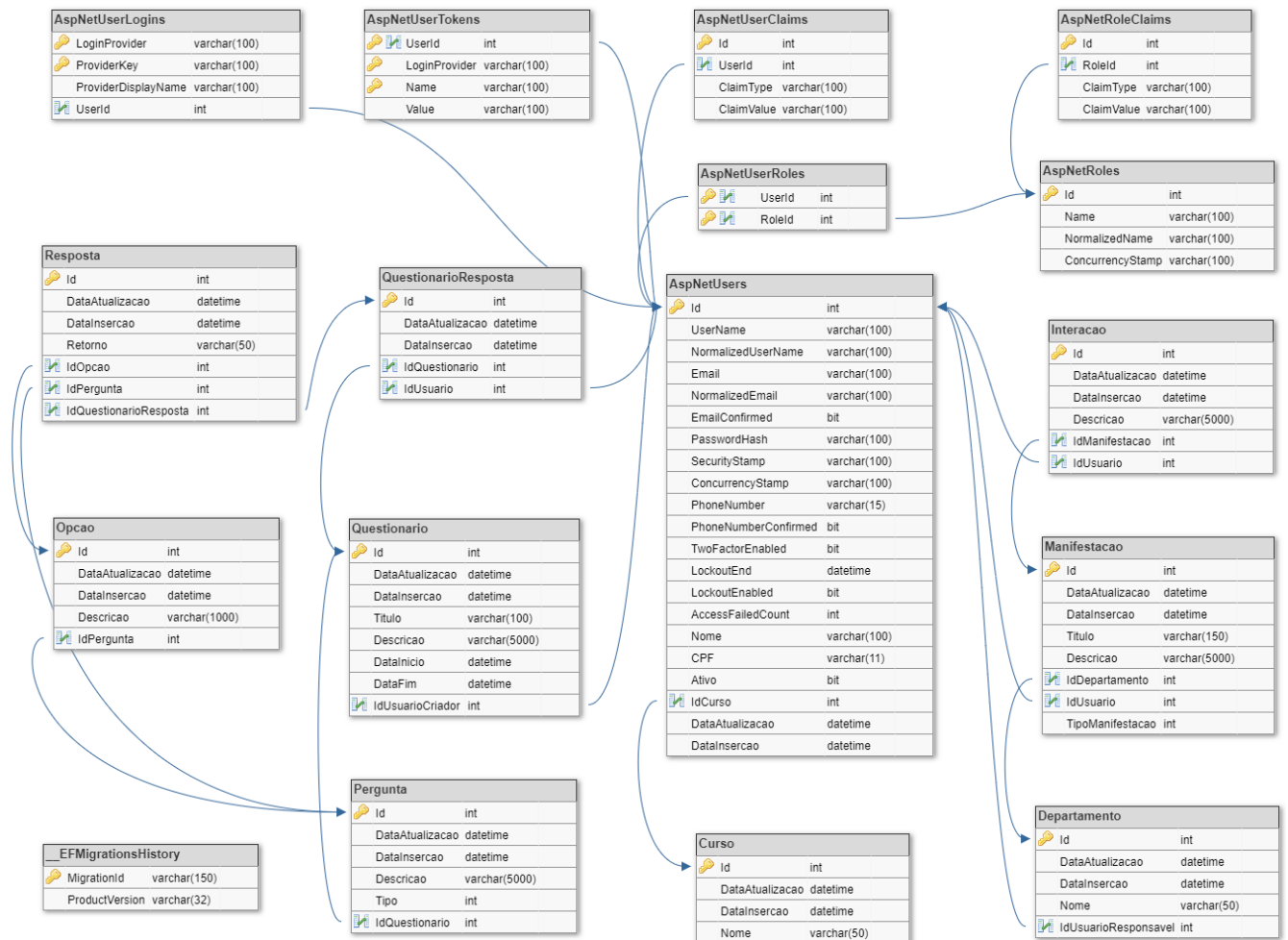
- *Identity*:

Responsável por tratar de toda a configuração da biblioteca *Identity* do ASP.NET Core, a qual tem a função de gerenciar toda a parte de segurança

relacionada à autenticação do usuário, incluindo login, cadastro e controle de sessão.

Com o software estruturado, pode-se extrair então seu modelo lógico que, para Cougo (1997, p. 7), é a representação abstrata e simplificada de um sistema real, com a qual se pode explicar ou testar o seu comportamento, em seu todo ou em partes. Logo, o modelo não é o objeto real, mas algo que o representa, com maior ou menor fidelidade. Abaixo, na Figura 6, está ilustrado o modelo lógico obtido.

Figura 6 - Modelo lógico



Fonte: do autor (2019)

5 RESULTADOS

Como resultado final, obteve-se um software web multiplataforma que atingiu os objetivos propostos, com categorização das manifestações em cinco diferentes tipos, aplicação de questionários criados por usuários administradores, acompanhamento de manifestações por parte de usuários e da instituição, geração de relatórios e segurança, com o uso do protocolo HTTPS e da biblioteca do ASP.NET Core denominada *Identity*.

Ao longo desta seção, são apresentadas as principais telas do software acompanhadas de uma breve descrição, onde pode-se observar o desenvolvimento de um layout simples e intuitivo, utilizando as cores do logotipo e do site oficial da Faculdade como base (azul, preto e branco), criado com o intuito de assemelhar os dois sistemas e não criar um grande impacto visual nos usuários.

5.1 Tela Inicial

A primeira tela do software é a inicial, representada na Figura 7. Nela, há uma breve explicação sobre o próprio sistema de ouvidoria, com tópicos como: “O que é”, “O que atende” e “Informações importantes”, com o intuito de informar o usuário sobre pontos importantes para a utilização adequada do sistema assim que ele fizer o acesso ao site.

Além desses tópicos informativos e explicativos, há também seis botões listados logo abaixo deles, com ícones e descrição, representando as principais funcionalidades do sistema: manifestações e questionários. Para retratar as manifestações, foram disponibilizados cinco botões, cada um referenciando um tipo de manifestação disponível no sistema, de tal forma que, ao serem acionados, direcionam o usuário à página de introdução do tipo referenciado. Já para representar os questionários, um único botão foi disponibilizado, o qual direciona o usuário à página de listagem de questionários assim que acionado.

Nesta tela também constam componentes que estão presentes nas demais telas do sistema, que são as barras de navegação presentes no topo e na lateral esquerda da página, possuindo elementos de navegação que servem de apoio ao usuário, com o intuito de facilitar sua experiência e agilizar suas ações no sistema. Estes componentes se adequam ao usuário que está ou não logado, disponibilizando botões de acordo com o nível de permissão do mesmo. Pode-se observar também que, no componente de navegação lateral, há a imagem do prédio principal da Faculdade, criando uma harmonia com a identidade visual da Instituição.

Figura 7 - Tela inicial



Fonte: do autor (2019)

5.2 Login

A tela de login do usuário, representada na Figura 8, é onde o usuário já cadastrado realiza o acesso e se autentica no sistema. Seu conteúdo possui o logotipo da AMF e um formulário com dois campos, sendo o primeiro para o e-mail e o segundo para a senha, além de dois botões, sendo o primeiro com a funcionalidade de submeter o formulário, e o segundo para redirecionar o usuário à tela de cadastro de usuário caso o mesmo não possua um.

Figura 8 - Tela de login

The image shows a web application interface for a 'Ouvidoria' (Ombudsman) system. The top navigation bar is blue and contains the text 'Ouvidoria' on the left and a 'ACESSAR' button with a user icon on the right. A vertical sidebar on the left side of the page is also blue and contains the AMF logo, the name 'Antonio Meneghetti Faculdade', and two menu items: 'Página Inicial' with a home icon and 'Manifestações' with a speech bubble icon and a dropdown arrow. The main content area is white and features the AMF logo at the top, followed by the word 'Ouvidoria'. Below this, there is a login form consisting of two input fields: 'E-mail' and 'Senha'. A blue button labeled 'ENTRAR' is positioned below the input fields. At the bottom of the form, there is a link that reads 'NÃO POSSUO UMA CONTA'.

Fonte: do autor (2019)

5.3 Cadastro de Usuário

Nesta tela, o usuário realiza seu cadastro no sistema, ação necessária para que as demais funcionalidades da ouvidoria estejam disponíveis para uso. Representada na Figura 9, ela possui um formulário com sete diferentes campos, sendo eles para o usuário informar seu nome completo, e-mail, CPF, telefone, seu curso (caso seja aluno), sua senha e também um último campo para confirmação de senha. Abaixo do formulário há também dois botões: o primeiro tem a função de submeter o formulário e o segundo redireciona o usuário à tela anterior, de login.

Figura 9 - Tela de cadastro de usuário

Cadastre-se!

Nome *

E-mail *

CPF *

Telefone

Curso

Senha *

Confirmar Senha *

CADASTRAR

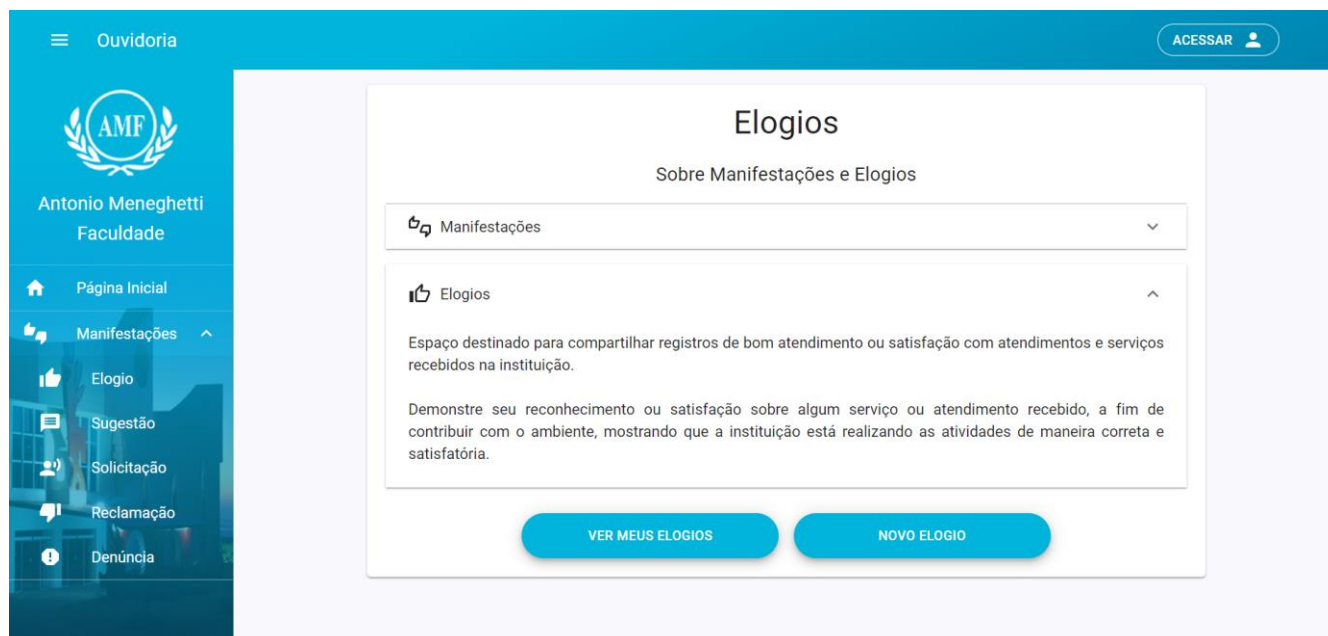
JÁ POSSUO UMA CONTA

Fonte: do autor (2019)

5.4 Introdução às Manifestações

As telas de introdução às manifestações, representadas, de maneira genérica, na Figura 10, são, na verdade, cinco diferentes telas, cada uma representando um dos tipos de manifestação disponíveis no sistema. Nelas, há uma introdução, instruções de uso e explicações de processos sobre as manifestações de modo geral e também sobre o tipo específico que cada uma das telas está tratando. Abaixo desses tópicos, as telas possuem dois botões: o primeiro redireciona o usuário à lista das manifestações e o segundo o redireciona à página de nova manifestação, ambos referenciando o tipo de manifestação que ele está acessando.

Figura 10 - Tela de introdução às manifestações



Fonte: do autor (2019)

5.5 Lista de Manifestações

Nestas telas, são listadas todas as manifestações já registradas pelo próprio usuário. Da mesma forma que a anterior, são cinco páginas distintas, cada uma referenciando um tipo de manifestação diferente. Nelas, há um item para cada manifestação registrada e, neste item, há uma breve introdução às informações que foram tratadas na referente manifestação, como título, descrição, data e horário, departamento, quantidade de manifestações e o último usuário a interagir com a mesma. Neste item há também um botão que, ao ser acionado, redireciona à página de visualização da respectiva manifestação.

A Figura 11 representa todos os elogios relatados por um usuário. Nela, encontra-se um único registro, exibindo seus detalhes como a data e o horário da sua criação (11/11/2019 01:03), o departamento ao qual a manifestação se dirige (Ouvidoria), o usuário criador (Usuário Exemplo), a quantidade de interações (uma) e o último usuário a interagir (Rhuan Matheus Drescher).

Figura 11 - Tela de listagem de manifestações



Fonte: do autor (2019)

5.6 Nova Manifestação

As telas de criação de manifestação, representadas, também de forma genérica, por meio da Figura 12, foram desenvolvidas seguindo o mesmo padrão das duas telas anteriores. Nelas, há um formulário contendo três campos, onde o primeiro refere-se ao título da manifestação que está sendo criada, o segundo representa o departamento ao qual o usuário está se manifestando e o último campo refere-se à descrição da manifestação, contendo todo o detalhamento do relato.

Figura 12 - Tela de nova manifestação

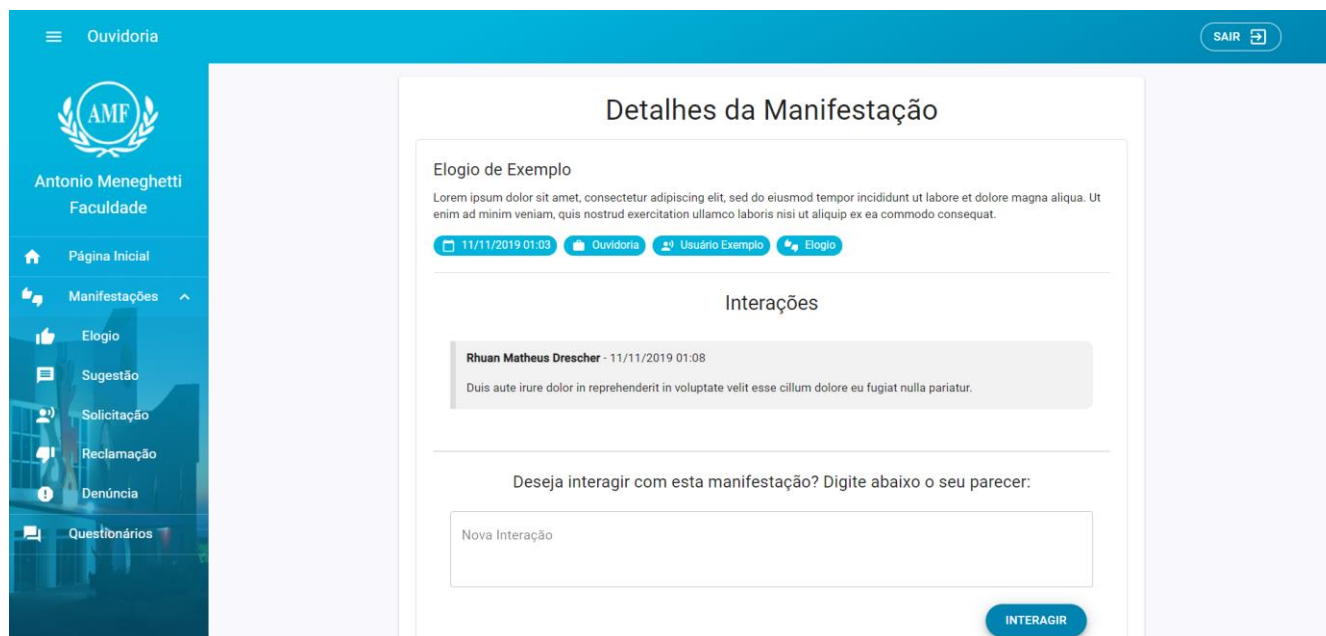
The image shows a web application interface for a 'Ouvidoria' (Ombudsman) system. The header is blue with the text 'Ouvidoria' and a 'SAIR' button. A sidebar on the left contains the AMF logo and the name 'Antonio Meneghetti Faculdade', along with navigation options: 'Página Inicial', 'Manifestações', 'Elogio', 'Sugestão', 'Solicitação', 'Reclamação', 'Denúncia', and 'Questionários'. The main content area is titled 'Novo Elogio' and contains a form with three fields: 'Título *', 'Departamento *' (with a dropdown menu showing 'Administrativo'), and 'Descrição *'. A blue 'SALVAR' button is located at the bottom right of the form.

Fonte: do autor (2019)

5.7 Visualização de Manifestação

Nesta tela, são exibidas todas as informações da manifestação selecionada. Nela, também se encontram todas as suas interações, sejam elas de administradores, do responsável pelo departamento ao qual a manifestação foi cadastrada ou então do próprio autor. Abaixo desta lista, há ainda um pequeno formulário destinado ao cadastro de interações com a referida manifestação. Este último, conta com apenas um campo e um botão de submissão, conforme ilustra a Figura 13, onde há os detalhes de um elogio cadastrado e suas interações.

Figura 13 - Tela de visualização de manifestação

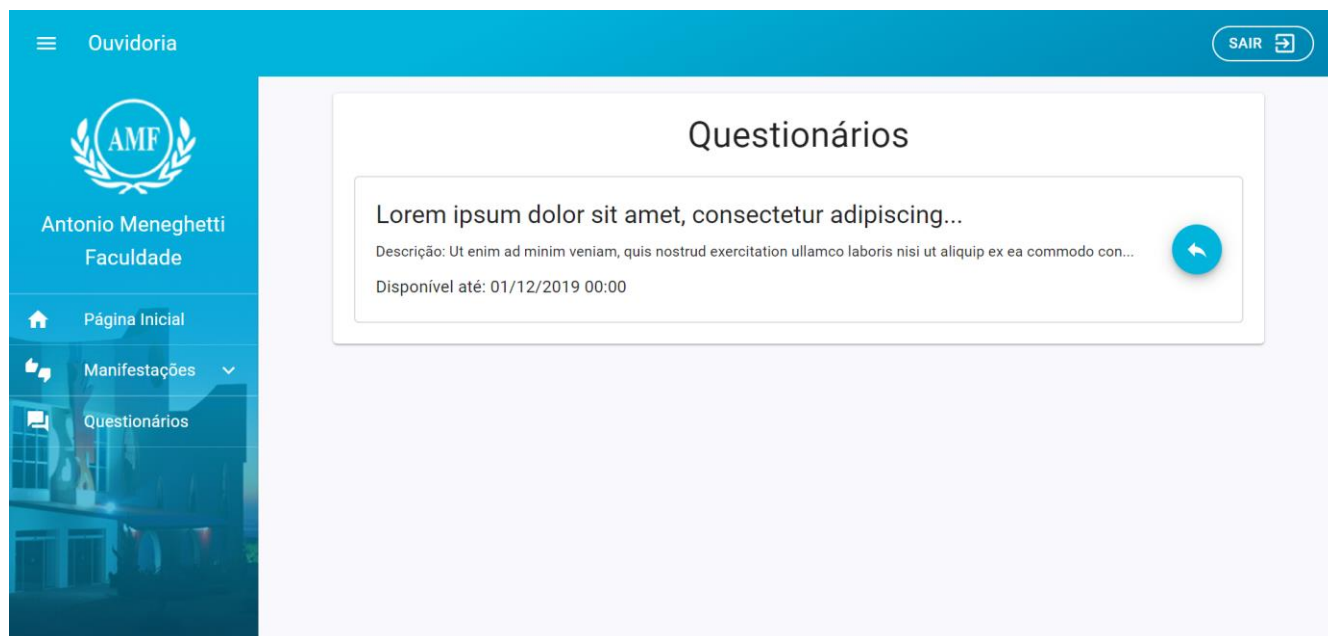


Fonte: do autor (2019)

5.8 Lista de Questionários

Nesta tela, representada na Figura 14, há a listagem de todos os questionários disponíveis para serem respondidos pelo usuário. Nela, ficam disponíveis os questionários que o usuário ainda não respondeu e que ainda estão em um período válido para serem respondidos. Em cada item da lista, há um breve detalhamento do questionário, como um trecho do título e da descrição do questionário, a data e o horário limite para responder ao questionário, e também um botão que, ao ser acionado, redireciona o usuário à tela de responder questionários.

Figura 14 - Tela de listagem de questionários



Fonte: do autor (2019)

5.9 Responder Questionário

A tela de responder questionários contém um formulário que lista todas as perguntas do questionário selecionado, com campos de acordo com o tipo de cada pergunta, de tal forma que, caso a pergunta seja descritiva, um campo do tipo texto é disponibilizado, e, caso a pergunta seja objetiva, um campo de seleção única é apresentado para cada opção da pergunta. Ao final do formulário, há um botão para submissão da resposta, conforme Figura 15.

Figura 15 - Tela de responder questionário

Menu: Ouvidoria SAIR

Antonio Meneghetti Faculdade

Página Inicial Manifestações Questionários

Responder questionário

Título: Lorem ipsum dolor sit amet, consectetur adipiscing elit

Descrição: Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur

1 - Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Resposta

2 - Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem Ipsum

ENVIAR RESPOSTA

Fonte: do autor (2019)

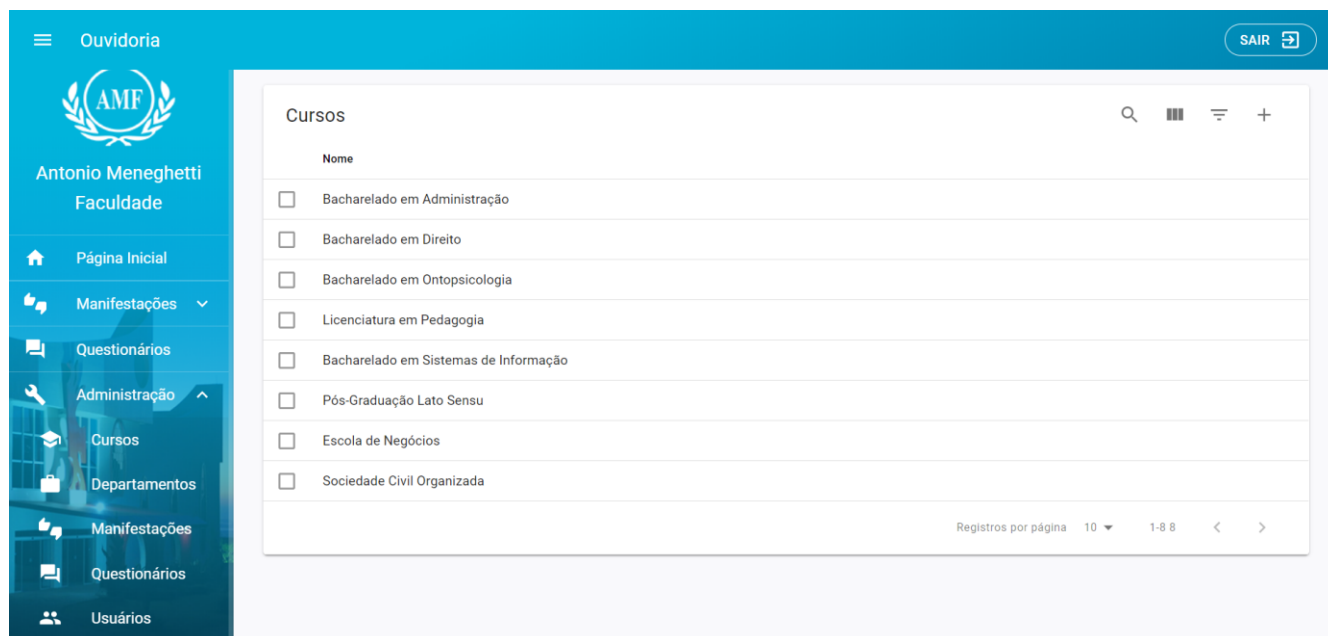
5.10 Telas Administrativas

As telas abaixo são páginas de acesso restrito a usuários administradores, de tal forma que todas aquelas que possuem funcionalidades de listagem de dados têm esses dados exibidos em uma tabela com diversas funcionalidades, como filtragem e paginação dos dados, pesquisa através de texto e também a opção de ocultar colunas que o usuário desejar.

5.10.1 Cursos e Departamentos

As telas de cursos e departamentos são duas telas que listam todos os cursos e departamentos existentes no sistema, tendo a opção de editar ou excluir um item, selecionando um registro na lista, ou então cadastrar um novo item, clicando no botão de adicionar. Todas essas três ações são controladas em formulários exibidos em uma modal que é aberta de acordo com a opção selecionada. A Figura 16 retrata a tela de cursos.

Figura 16 - Tela administrativa de cursos

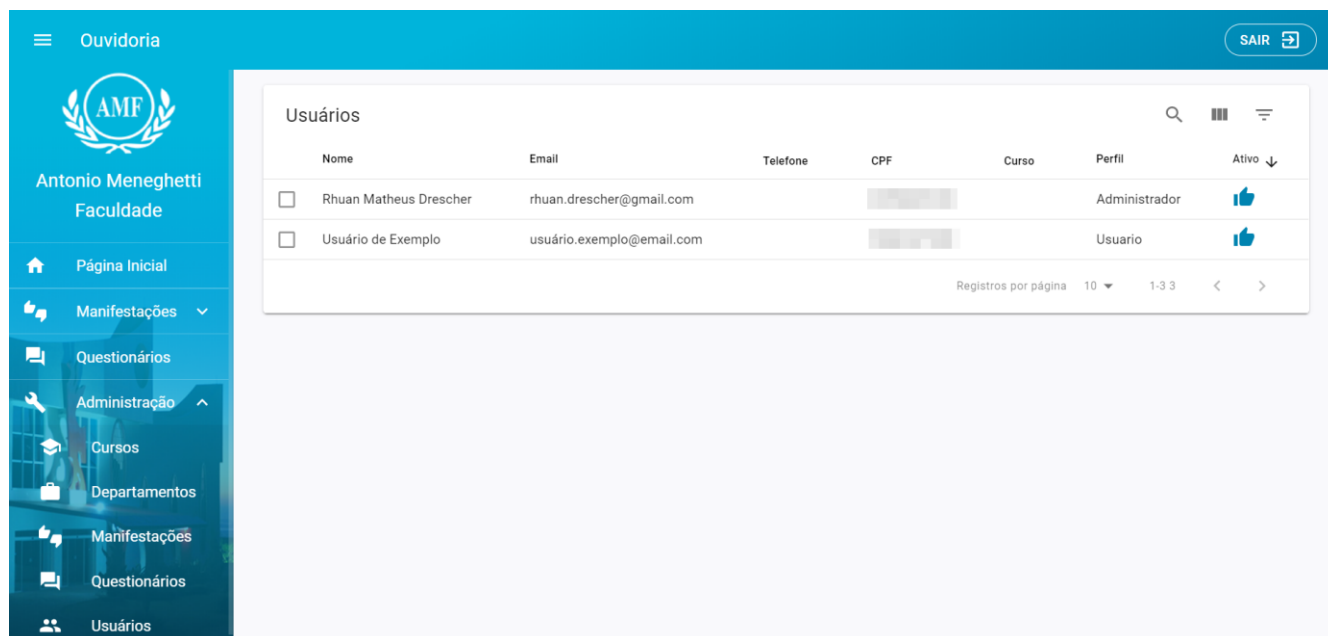


Fonte: do autor (2019)

5.10.2 Usuários

Na tela de usuário, exibida na Figura 17, são listados todos os usuários que possuem cadastro no sistema, juntamente de suas informações. Nesta tela, diferente das duas anteriores, há apenas a opção de edição, onde ficam disponíveis para alteração apenas algumas informações do usuário, como nome, telefone, curso, o nível de permissão e a opção de ativar ou inativar o acesso, funcionando também como uma modal.

Figura 17 - Tela administrativa de usuários



Fonte: do autor (2019)

5.10.3 Questionários

A tela de questionário é responsável por listar todos os questionários existentes no sistema com uma breve exibição de suas informações, como um trecho do título e da descrição, as datas de início e fim do período de disponibilidade do questionário, a quantidade de perguntas e o número de respostas. Nesta lista, também estão disponíveis três botões, sendo o primeiro para redirecionar a uma tela de visualização das suas respostas de modo separado por usuário, o segundo para redirecionar a tela de pré-visualização, de modo a visualizar a forma que o questionário foi elaborado, com suas perguntas e opções, e o último botão servindo para redirecionar o usuário à tela de relatório de questionários.

Conforme Figura 18, ela também disponibiliza um botão para cadastrar um novo questionário, tal que, ao ser acionado, redireciona o usuário à tela de cadastro de questionário.

Figura 18 - Tela administrativa de questionários

Lista de Questionários								
Id	Título	Descrição	Início	Fim	Criador	Perguntas	Respostas	Ações
9	Lorem ipsum dolor si...	Ut enim ad minim ven...	10/11/2019 00:00	01/12/2019 00:00	Rhuan Matheus Drescher	2	0	

Registros por página 10 1-9 9 < >

Fonte: do autor (2019)

5.10.4 Cadastrar Questionário

A tela de cadastro de questionários, mostrada na Figura 19, disponibiliza um formulário dinâmico, onde o usuário preenche informações como o título, a descrição, as datas de início e de término do período de respostas do questionário, bem como as perguntas e suas opções. Nele, ficam disponíveis botões para adicionar ou remover perguntas e opções, fazendo com que o usuário escolha suas quantidades de maneira prática. Ao final do formulário há um último botão, responsável por submeter o formulário.

Figura 19 - Tela administrativa de cadastrar questionário

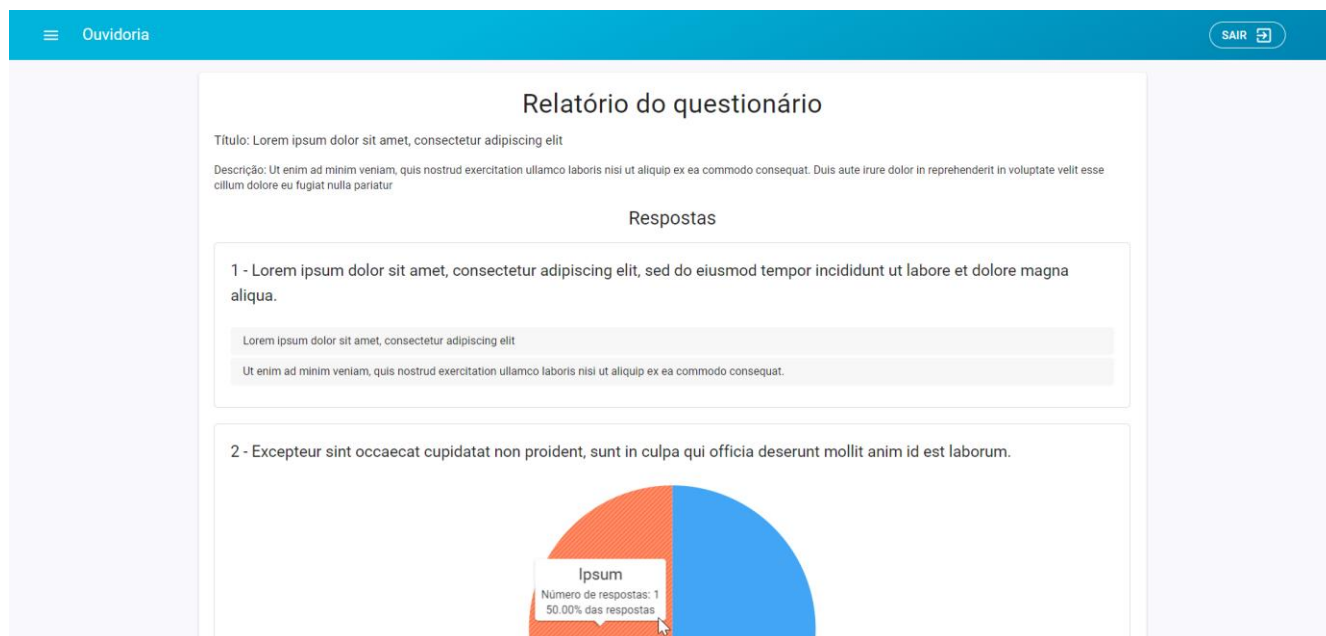
The screenshot shows a web interface for creating a new questionnaire. At the top, there is a blue header with the text 'Ouvidoria' and a 'SAIR' button. The main content area is titled 'Novo Questionário' and contains several input fields: 'Título', 'Descrição', 'Data inicial' (set to 11/11/2019 00:03), and 'Data final' (set to 11/11/2019 00:03). Below these fields is a section titled 'Perguntas' with a 'NOVA PERGUNTA' button. Under 'Perguntas', there is a sub-section 'Pergunta nº 1' with a 'Descrição' field and a 'Tipo da Pergunta' dropdown menu currently set to 'Dissertativa'.

Fonte: do autor (2019)

5.10.5 Relatório de Questionários

Nesta tela, representada na Figura 20, os administradores podem visualizar um relatório a respeito de todas as respostas obtidas em um questionário. Nela, são exibidas todas as perguntas do questionário selecionado, juntamente com suas respostas. A exibição das respostas, entretanto, pode se dar de forma diferente, a depender do tipo de pergunta: no caso de perguntas dissertativas, as respostas dos usuários são exibidas uma abaixo da outra; já no caso de questionamentos objetivos, um gráfico de pizza é exibido, informando a quantidade de escolhas para cada opção e sua percentagem em relação às demais.

Figura 20 - Tela administrativa de relatório de questionários



Fonte: do autor (2019)

5.10.6 Manifestações

A tela de manifestações apresenta a lista de todas as manifestações realizadas pelos usuários. Exibida na Figura 21, ela possui informações como o título e o tipo da manifestação, o usuário criador, a data e horário da criação, o número de manifestações e o último usuário a interagir. Ao final de cada item, um botão é disponibilizado com a função de direcionar o usuário à página de visualização da referente manifestação.

Figura 21 - Tela administrativa de manifestações

The screenshot displays the administrative interface for 'Ouvidoria' (Ombudsman) at 'Antonio Meneghetti Faculdade'. The interface is divided into a sidebar and a main content area.

Sidebar (Left):

- Logo: AMF (Antonio Meneghetti Faculdade)
- Navigation menu:
 - Página Inicial
 - Manifestações
 - Questionários
 - Administração
 - Cursos
 - Departamentos
 - Manifestações
 - Questionários
 - Usuários

Main Content Area (Right):

Lista de Manifestações

Id	Título	Tipo	Usuário	Criado em	Interações	Última Interação	Ações
1	Elogio de Exemplo	Elogio	Rhuan Matheus Drescher	17/11/2019 21:28	0		

Registros por página: 10 | 1-1 1

Fonte: do autor (2019)

6 DISCUSSÃO DOS RESULTADOS

Com o intuito de coletar a opinião do público alvo e avaliar o resultado final, foram aplicados dois questionários distintos aos alunos e professores da Instituição, utilizando a ferramenta de questionários do próprio software. Para que o público pudesse acessar o software, foram utilizadas algumas ferramentas gratuitas do Azure, um conjunto de serviços de nuvem da Microsoft, como a hospedagem do software e da base de dados.

Após o sistema estar hospedado e disponível para acesso, a Faculdade enviou um comunicado divulgando o software entre seus professores e alunos da graduação, através de um e-mail, informando que haviam dois questionários disponíveis para acesso. Este e-mail pode ser visualizado na Figura 22.

Figura 22 - E-mail enviado pela AMF para divulgação do software e seus questionários



Fonte: do autor (2019)

Os questionários ficaram disponíveis até o dia 18/11/2019 onde, ao seu término, obteve-se um total de quarenta e nove respostas. Estas respostas podem ser analisadas nos itens abaixo:

6.1 Questionário de escala de usabilidade do sistema

O questionário de usabilidade do sistema (SUS), segundo (JORDAN, THOMAS, *et al.*, 1996), foi criado em 1986 por John Brooke, com o intuito de apresentar uma escala baseada na efetividade, eficiência e satisfação do software em relação a seus usuários através de dez perguntas objetivas, onde nelas ficam disponibilizadas cinco opções de escolha, que vão do número um ao cinco, sendo o primeiro entendido como “discordo totalmente” e o segundo como “concordo totalmente”. Para analisar os resultados, é necessário realizar um cálculo de acordo com o número da pergunta e o valor da resposta: primeiramente é obtido o valor de cada resposta, de acordo com o número da pergunta, onde para perguntas de número ímpar é subtraído o número um do valor da resposta e, para perguntas de número par, é subtraído o valor da resposta do número cinco. Ao final do cálculo do valor das respostas, é realizada a soma entre eles e, por fim, multiplicada por 2,5. Como resultado, obtém-se um número entre zero e cem, que são medidos na escala representada na Figura 23.

Figura 23 - Quadro de pontuação do SUS

Pontuação SUS	Usabilidade
90,9	Melhor Imaginável
85,5	Excelente
71,4	Bom
50,9	Indiferente
35,7	Pobre
20,3	Horrível
12,5	Pior Imaginável

Fonte: (MOULDEN, 2018, tradução nossa)

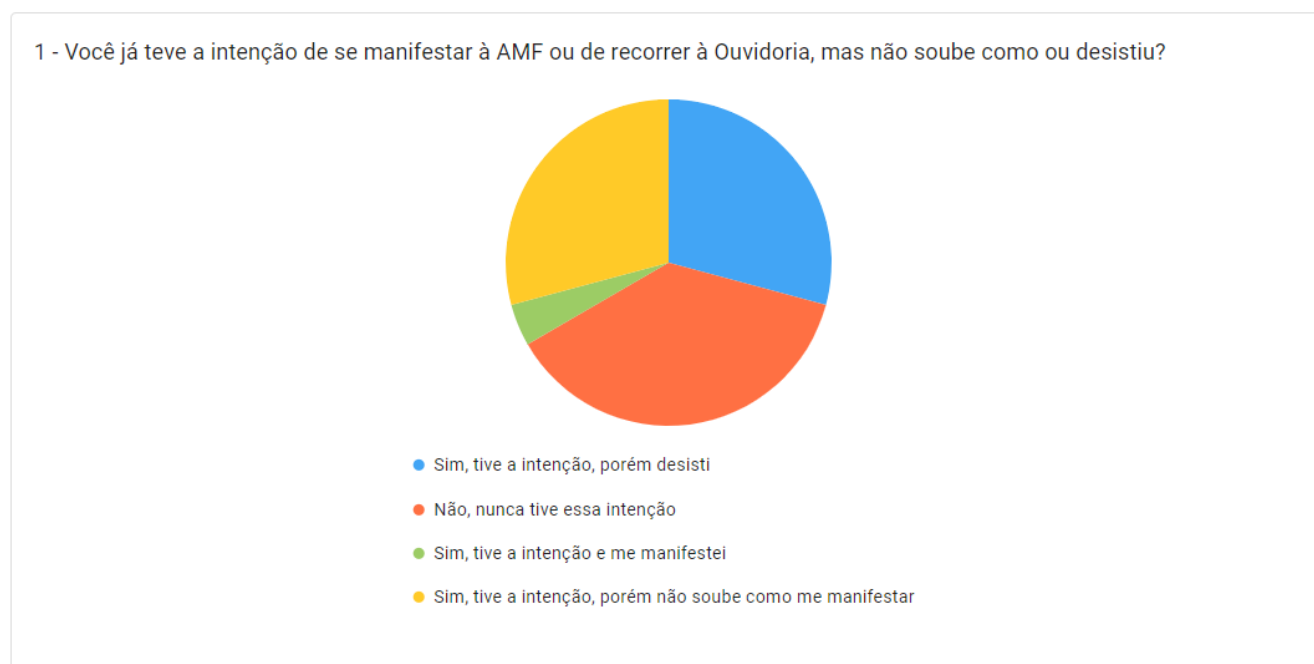
Para medir a usabilidade deste sistema, o autor utilizou dez perguntas, as quais estão disponíveis no APÊNCIDE A, onde obteve-se vinte e cinco respostas e que, ao final do cálculo, obteve-se o valor 91,30, caracterizando a usabilidade do software, então, como melhor imaginável e que, segundo Gonçalves (2009), com uma boa usabilidade, um software cria um impacto positivo na realização de tarefas no sistema em relação à eficiência, eficácia e produtividade, permitindo que o usuário atinja seus objetivos com mais satisfação e menos esforço.

6.2 Questionário de viabilidade do software

O questionário de viabilidade do software, representado no APÊNDICE B, foi elaborado pelo autor com o intuito de analisar a viabilidade da utilização do software como sistema de ouvidoria oficial da Instituição. Ele contém seis perguntas objetivas, com o intuito de avaliar o atual sistema e também compará-lo com este software, e uma dissertativa, com o objetivo de coletar opiniões gerais sobre o sistema. No total, vinte e quatro respostas foram obtidas, as quais suas análises podem ser observadas abaixo.

Conforme ilustrado na Figura 24, a maioria dos respondentes já teve a intenção de se manifestar à Instituição, sendo somente 37,5% respostas negativas. Percebe-se também que, dentre os que quiseram se manifestar, somente uma realizou sua manifestação, remetendo ao fato de que na maioria das vezes em que uma pessoa tem a intenção de se manifestar, ela acaba não se manifestando, seja por desistência ou porque não tem o conhecimento de como fazê-la, o que salienta o número de manifestantes que a AMF deixa de ouvir com o uso do sistema de urnas.

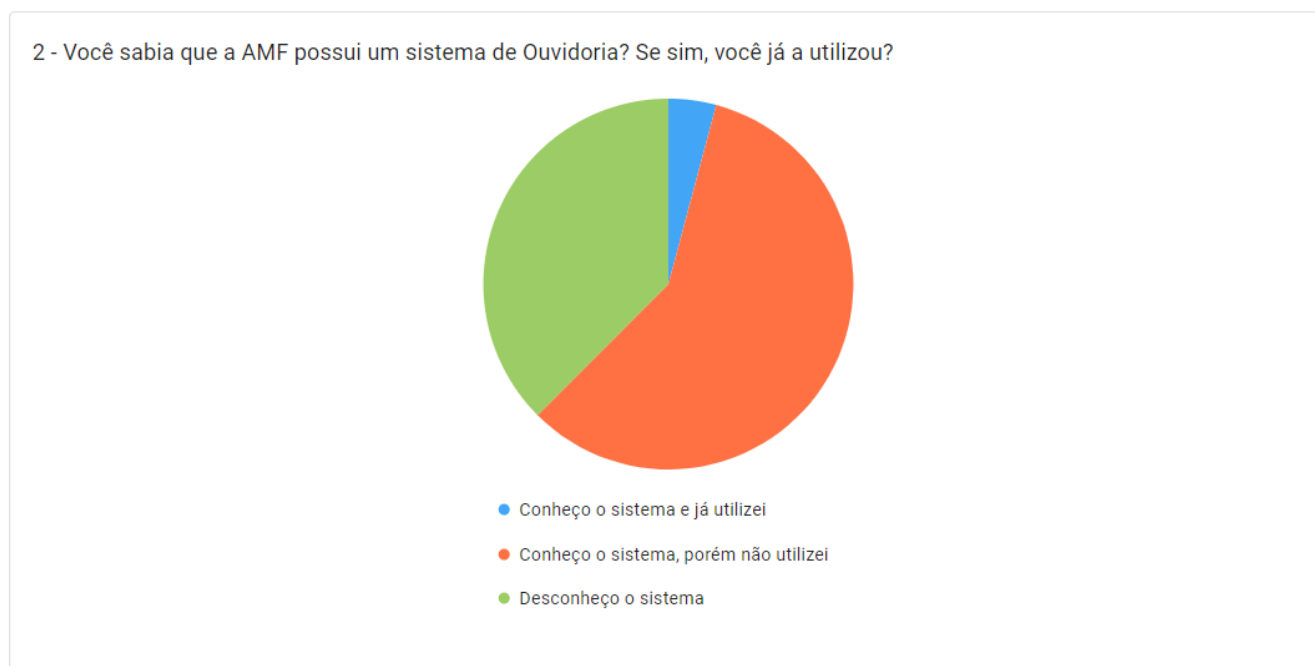
Figura 24 - Respostas da pergunta 1



Fonte: do autor (2019)

Conforme a Figura 25, a maioria dos respondentes tem conhecimento da existência do atual sistema de ouvidoria da Faculdade, porém somente uma se manifestou. Percebe-se também que o número de respondentes que não conhecem o sistema também é notável, sendo 37,5% do total.

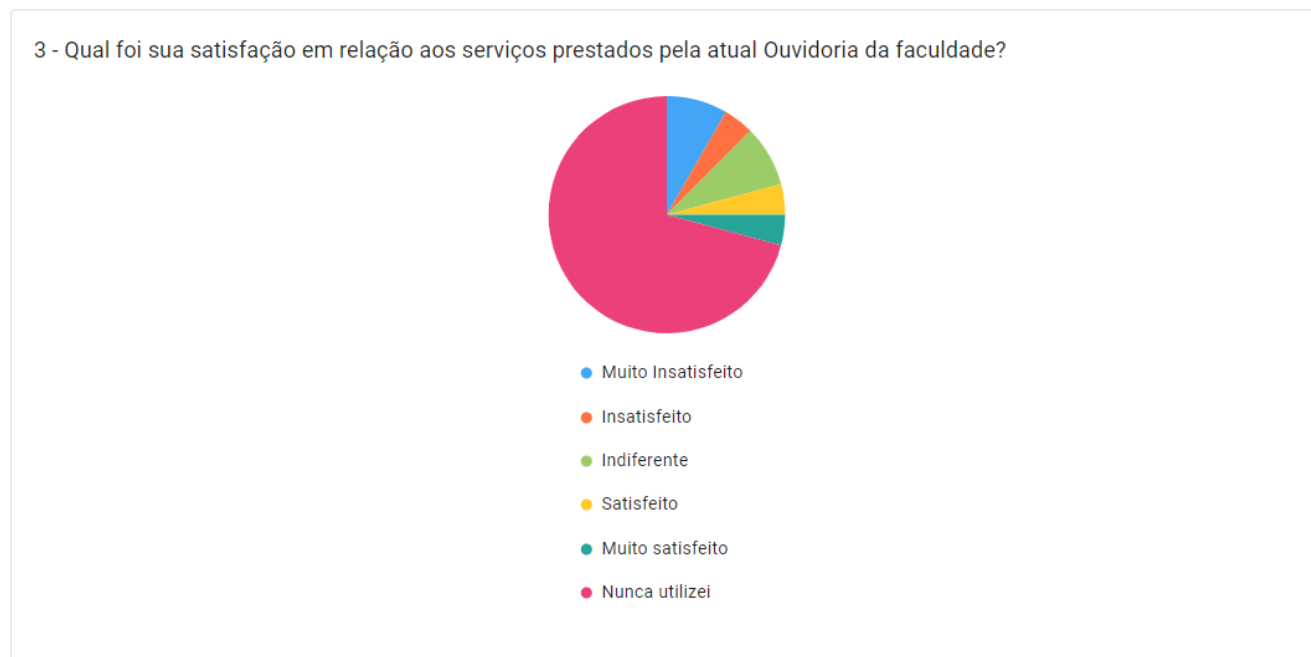
Figura 25 - Respostas da pergunta 2



Fonte: do autor (2019)

Segundo a Figura 26, 70,83% dos respondentes não utilizaram o sistema de urnas da Faculdade, porém, se comparado às respostas das perguntas anteriores, essas informações não são compatíveis. Isso pode remeter ao fato de que, mesmo não utilizando o sistema, as pessoas possam ter uma opinião sobre ele, através de pessoas que já utilizaram e que conhecem o atendimento do sistema de urna. Dentre esses, o único respondente que utilizou o sistema, escolheu a opção “Muito insatisfeito”. Os demais, na sua maioria, mesmo sem utilizarem o sistema, não têm uma opinião positiva sobre o atual sistema.

Figura 26 - Respostas da pergunta 3



Fonte: do autor (2019)

Conforme a Figura 27, a pergunta de número 4 obteve 100% das respostas na opção “Agregará muito valor”, evidenciando que, mesmo para um grande número de respondentes que desconheciam o sistema de urnas da Faculdade, o software possui pontos positivos e que são de real necessidade da Instituição, trazendo benefícios através de suas facilitadas e seguras funcionalidades.

Também evidencia que o software só tem a agregar valor à Instituição, visto que o mesmo disponibilizará um meio de comunicação mais seguro, eficiente e prático entre a Faculdade e seu público alvo, o qual reconhece o esforço da AMF em relação à preocupação em ouvi-los sobre seu próprio ambiente de convívio.

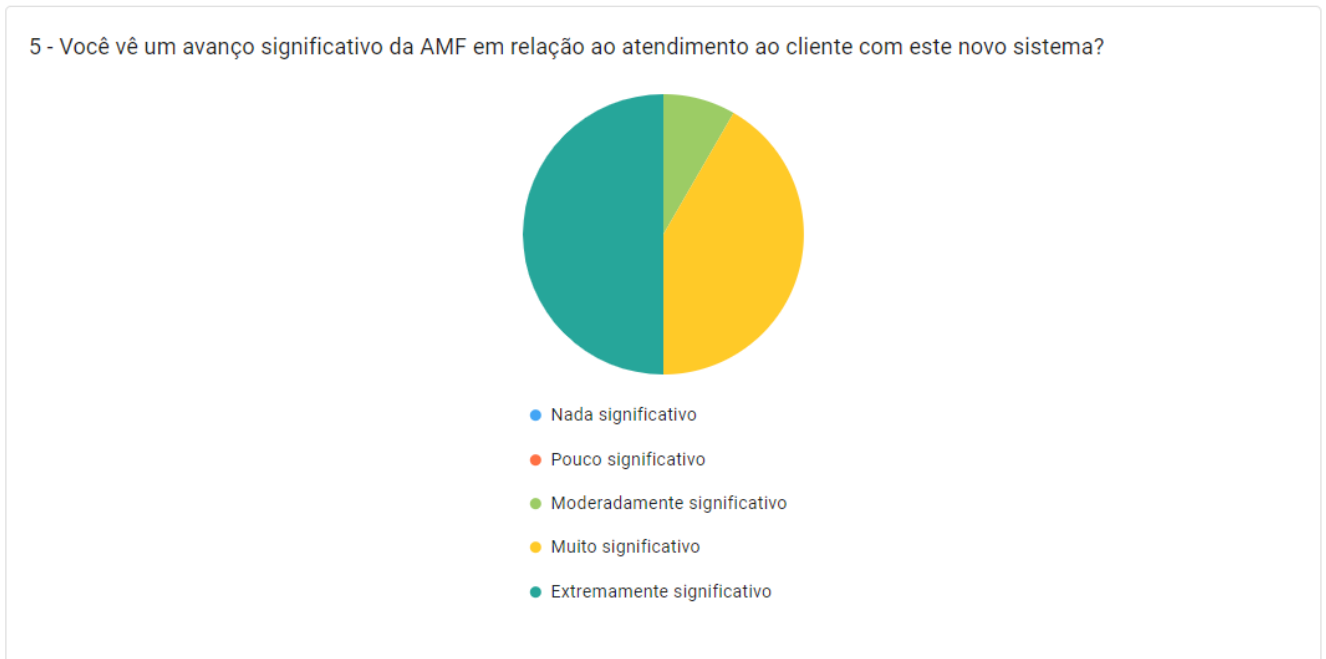
Figura 27 - Respostas da pergunta 4



Fonte: do autor (2019)

A Figura 28 representa as respostas da pergunta 5 onde, em sua maioria, foram positivas, evidenciando que o atendimento da AMF pode se tornar ainda melhor através do uso desse software e que a Instituição se mostra mais preocupada em tornar o meio pelo qual ouve seu público alvo mais prático, seguro e eficiente, através de um sistema web de ouvidoria.

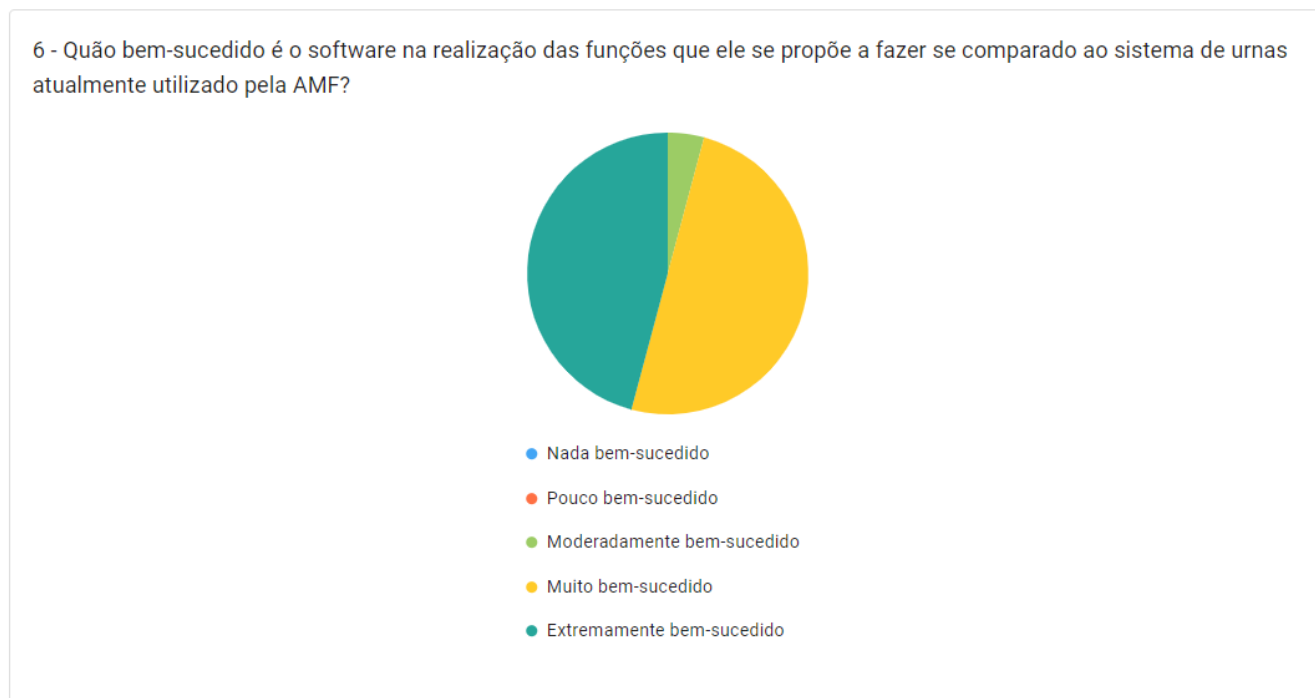
Figura 28 - Respostas da pergunta 5



Fonte: do autor (2019)

Conforme a Figura 29, novamente a maioria das respostas foram positivas, porém agora relacionado à realização das funções do software comparado com o sistema atual, o de urnas. Isso deixa claro o significativo avanço na realização das tarefas do sistema, evidenciando a evolução na praticidade, rapidez e segurança em relação ao atual, de urnas.

Figura 29 - Respostas da pergunta 6



Fonte: do autor (2019)

A Figura 30 representa algumas das opiniões e comentários relatados na pergunta 7, onde pode-se observar vários elogios ao software, o reconhecimento da necessidade de implantação do mesmo na Instituição e também algumas sugestões de novas funcionalidades. Respostas como estas, demonstram o reconhecimento do público em relação à necessidade de um sistema de ouvidoria mais moderno na Faculdade, onde há, inclusive, a participação do público na elaboração de novas ideias para o software, evidenciando ainda mais sua preocupação com a Instituição e também seu engajamento em relação a esta ideia.

Figura 30 - Respostas da pergunta 7

7 - Deixe abaixo alguma recomendação ou comentário referente a esta ideia.

Consegue oferecer um canal simples e prático de comunicação entre o aluno e a faculdade.

Acredito que automatização do sistema de Ouvidoria, irá facilitar aos envolvidos com a Faculdade (funcionários, alunos, professores, etc), expressarem a sua opinião de forma mais confortável, permitindo ser realizado de qualquer lugar e ainda, será positivo para a administração da própria instituição que receberá as manifestações de forma instantânea e mais organizada do que o atual sistema de urna.

A ampliação do sistema de ouvidoria para o acesso digital facilitará bastante a comunicação entre os alunos e instituição referente à reclamações, dúvidas e sugestões.

Ideia genial, parabéns pela iniciativa!

Muito bom, principalmente coletar informações das pessoas e ver qual é a visão de cada um sobre o sistema e sobre a AMF.

A seguir pode ser implementada a função de manifestação anônima. Na minha experiência aqui, esta é a única vantagem que o sistema de urnas ainda tem sobre este software. Em geral, todos os envolvidos estão de parabéns, tanto quem desenvolveu e se preocupou em deixar o sistema seguro, prático e moderno, quanto a Instituição em sua preocupação por melhorar sua relação com seu público.

Ótima ideia!

Está ótimo. Avante!

Nunca utilizei os serviços prestador pelo sistema de ouvidoria da AMF, mas julgo necessário para que se um dia eu necessite ou qualquer pessoa que vá precisar ter um sistema fácil e preciso para se utilizar

Todo o sistema de ouvidoria necessita dar feedback rápido e conclusivo ao solicitante/cliente.

O software propõe-se a solucionar um problema evidente dentro da Instituição: a falta de um sistema de ouvidoria confiável, atrativo e de fácil utilização. Nesse sentido, é muito bom ver os alunos da AMF olhando de maneira crítica para dentro da Faculdade e se colocando em movimento para a construção de soluções viáveis e de qualidade! O software cumpre de maneira extremamente satisfatória com seus objetivos! Parabéns pelo trabalho!

O sistema proposto poderia ter uma forma em que o interessado possa agendar uma consulta/conversa presencial, para poder apresentar assunto que julgue sigiloso ou que não queira deixar registrado no sistema, ou, ainda, o próprio interessado sinta-se desprotegido para fazê-lo aqui, por ser assunto pessoal ou íntimo. Com isso, creio que ficam atendidos todos os tipos de assuntos e formas de atendimento que uma boa ouvidoria deve ter. A final, somos uma escola de Ontopsicologia. Obrigado e parabéns pelo trabalho.

Será de grande valia, pois esse software irá condizer com a real atualidade em que vivemos atualmente com relação à tecnologia e usar uma urna como forma de ouvidoria para uma faculdade com esta foge muito da qualidade que a mesma preza.

O sistema é excelente!!!

Sistema interessante que vai agregar valor e remover a necessidade da urna

Fonte: do autor (2019)

7 CONSIDERAÇÕES FINAIS

Este trabalho foi desenvolvido com o intuito de criar um sistema de ouvidoria mais rápido, prático e seguro à Antonio Meneghetti Faculdade, possibilitando à Instituição melhorar o seu atendimento ao seu público alvo. Conforme demonstrado, a Faculdade utiliza um sistema de urnas como sua ouvidoria, um canal que não é prático e seguro, o que pode ocasionar na não utilização do mesmo pelo público alvo da AMF.

Buscando resolver problemas como esses, foi desenvolvido um software web que possibilitou a categorização de manifestações em cinco tipos distintos, permitiu usuários, responsáveis e administradores a acompanharem o andamento das manifestações e interagir com as mesmas e também disponibilizou um sistema de questionários, onde a Instituição pode aplicar questionários em determinados períodos com perguntas a sua escolha.

Após o desenvolvimento ser concluído, o software foi publicado, utilizando serviços em nuvem, e divulgado pela Instituição, com o intuito de recolher opiniões de alunos e professores da graduação sobre o sistema web de ouvidoria.

Com esses dados, pode-se notar que a Faculdade necessita do sistema, na visão dos respondentes, e que o mesmo foi bem visto pelos mesmos, tanto na realização de suas atividades em relação ao sistema de urnas quanto na sua usabilidade.

Desse modo, conclui-se que ações como essa, de um desenvolvimento de um software com finalidades de melhorar serviços já prestados ou não pela Faculdade, agregam valor à Instituição, de modo que facilite ainda mais seus processos em relação ao seu público alvo.

Como trabalhos futuros, fica a disponibilidade da construção de soluções de *Business Intelligence* e *Analytics* pela Faculdade, onde poderá ser feita a análise aprofundada dos dados ali tratados.

8 REFERÊNCIAS

ALLET, A. Introdução ao REST WS. **DevMedia**, 2013. Disponível em: <<http://bit.ly/RESTIntroduction>>. Acesso em: 11 nov. 2019.

BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. **UML: guia do usuário**. 6ª. ed. Rio de Janeiro: Elsevier, 2005.

CGU, M. D. T. E. C.-G. D. U. Manual de Ouvidoria Pública. **Manual de Ouvidoria Pública**, Brasília, nov. 2018. Disponível em: <<http://bit.ly/ManualOuvidoria>>. Acesso em: 30 set. 2019.

COUGO, P. **Modelagem Conceitual e Projeto de Bancos de Dados**. 1ª. ed. Rio de Janeiro: Campus, 1997.

DB-ENGINES. Db-Engines. **DB-Engines Ranking - Trend Popularity**, 2019. Disponível em: <<http://bit.ly/DbEngines>>. Acesso em: 17 nov. 2019.

DIONISIO, E. J. Introdução ao TypeScript. **DevMedia**, 2016. Disponível em: <<http://bit.ly/TypescriptIntroduction>>. Acesso em: 17 jun. 2019.

DIONISIO, E. J. Introdução ao Visual Studio Code. **DevMedia**, 2016. Disponível em: <<http://bit.ly/VisualStudioCodeIntroduction>>. Acesso em: 18 jun. 2019.

DOS SANTOS, A. M. et al. Diretrizes para a construção de casos de uso eficazes. **Revista de Informática Aplicada**, Santo André, v. 2, p. 71, jun. 2006.

ECMA. C# Language Specification. **ECMA**, 2017. Disponível em: <<http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-334.pdf>>. Acesso em: 23 set. 2019.

EVEN3. Metodologia Científica: guia simplificado para escrever a sua. **Even3 Blog**, 2017. Disponível em: <<http://bit.ly/HowToScientificMethodology>>. Acesso em: 30 set. 2019.

GASPAROTTO, H. M. Os 4 pilares da programação orientada a objetos. **DevMedia**, 2014. Disponível em: <<http://bit.ly/POOPillars>>. Acesso em: 24 set. 2019.

GÊA, R. A Importância da Ouvidoria para a Sociedade - 2ª Publicação. **Administradores**, 2011. Disponível em: <<https://bit.ly/2uOXUsF>>. Acesso em: 17 abr. 2019.

GIL, A. C. **Métodos e técnicas de pesquisa social**. 6^a. ed. São Paulo: Atlas S.A., 2008.

GITHUB. GitHub Octoverse. **The State of the Octoverse**, 2019. Disponível em: <<http://bit.ly/GitHubOctoverse>>. Acesso em: 17 nov. 2019.

GLÜER, L. M. A Ouvidoria Universitária como espaço de interlocução com os diferentes públicos, 5 set. 2005. 3.

GODOY, A. S. Introdução à pesquisa qualitativa e suas possibilidades. **Revista de Administração de Empresas**, São Paulo, v. 35, n. 2, p. 57-63, abr. 1995.

GONÇALVES, M. K. Usabilidade de software: estudo de recomendações básicas para verificação do nível de conhecimento dos alunos dos cursos de Design Gráfico e Sistemas de Informação da UNESP/Bauru. **Universidade Estadual Paulista**, Bauru, 2009. Disponível em: <<http://hdl.handle.net/11449/89684>>.

JORDAN, P. et al. **Usability Evaluation In Industry**. 1^a. ed. [S.l.]: CRC Press, 1996.

LOCAWEB. O que é Single Page Application. **Locaweb**, 2016. Disponível em: <<http://bit.ly/WhatIsSPA>>. Acesso em: 30 set. 2016.

LOTAR, A. **Como programar com ASP.NET e C#**. 2. ed. São Paulo: Novatec Editora, 2010.

LYCAN, R. Uma introdução ao TypeScript. **Rafaell Lycan**, 2017. Disponível em: <<http://bit.ly/ATypeScriptIntroduction>>. Acesso em: 23 set. 2019.

MARCONDES, A. R. Single Page Applications e outras maravilhas da web moderna. **iMasters**, 2013. Disponível em: <<http://bit.ly/SPAandOthers>>. Acesso em: 30 set. 2019.

MDN. O que é JavaScript? **Mozilla Developers Network**, 2019. Disponível em: <<http://bit.ly/WhatIsJavaScript>>. Acesso em: 30 set. 2019.

MICROSOFT. O histórico da linguagem C#. **Microsoft Docs**, 2017. Disponível em: <<http://bit.ly/CSharpVersionHistory>>. Acesso em: 17 jun. 2019.

MICROSOFT. Guia do.NET Core. **Microsoft Docs**, 2018. Disponível em: <<http://bit.ly/DotnetCore>>. Acesso em: 17 jun. 2019.

MICROSOFT. Introdução ao ASP.NET Core. **Microsoft Docs**, 2019. Disponível em: <<http://bit.ly/AspNetCore3>>. Acesso em: 30 set. 2019.

MICROSOFT. O que é o Azure Data Studio. **Microsoft Docs**, 2019. Disponível em: <<http://bit.ly/AzureDataStudio>>. Acesso em: 17 jun. 2019.

MICROSOFT. Um tour pela linguagem C#. **Microsoft Docs**, 2019. Disponível em: <<http://bit.ly/TourCSharp>>. Acesso em: 23 set. 2019.

MICROSOFT. What Is .NET? **Microsoft.NET**, 2019. Disponível em: <<http://bit.ly/WhatIsDotNet>>. Acesso em: 24 set. 2019.

MONQUEIRO, J. C. B. Programação Orientada a Objetos: uma introdução. **Hardware**, 2007. Disponível em: <<http://bit.ly/POOIntroduction>>. Acesso em: 24 set. 2019.

MOULDEN, B. The System Usability Scale. **Discida**, 2018. Disponível em: <<http://bit.ly/SystemUsabilityScale>>. Acesso em: 17 nov. 2019.

PIRES, E. DDD não é arquitetura em camadas. **Eduardo Pires**, 2016. Disponível em: <<http://bit.ly/DDDIsntAnArchitecture>>. Acesso em: 30 set. 2019.

RAMPINELLI, F. Ouvidoria: O que é e a sua importância para a sua empresa. **DDS**, 2015. Disponível em: <<http://bit.ly/ImportanciaOuvidoria>>. Acesso em: 21 jul. 2019.

REACT. Introduzindo JSX. **React**, 2017. Disponível em: <<http://bit.ly/JSXIntroduction>>. Acesso em: 30 set. 2019.

REACT. React. **React**, 2017. Disponível em: <<http://bit.ly/ReactJsSite>>. Acesso em: 30 set. 2019.

SCHIMIGUEL, J. Gerenciamento de Banco de Dados: Análise Comparativa de SGBD's. **DevMedia**, 2014. Disponível em: <<http://bit.ly/SGDBsComparison>>. Acesso em: 17 jun. 2019.

THIOLLENT, M. **Metodologia da pesquisa-ação**. 2. ed. São Paulo: Cortez, 1947.

TRENDS, G. Google Trends. **Google Trends**, 2019. Disponível em: <<http://bit.ly/JSFrameworksTrends>>. Acesso em: 17 nov. 2019.

APÊNCIDE A

Questionário de Usabilidade do sistema

Preview questionário

Título: Questionário de escala de usabilidade do sistema

Descrição: Este questionário tem como objetivo coletar opiniões de usabilidade do sistema, através de dez questões objetivas que consistem em avaliar se o quão de acordo o usuário está em relação à colocação. Todas as questões estão utilizando a escala Likert, que vai de 1 a 5, sendo 1 considerado como "Discordo completamente" e 5 "Concordo completamente".

1 - Eu acho que gostaria de usar esse sistema

1 2 3 4 5

2 - Eu acho o sistema desnecessariamente complexo.

1 2 3 4 5

3 - Eu achei o sistema fácil de usar.

1 2 3 4 5

4 - Eu acho que precisaria de ajuda de uma pessoa com conhecimentos técnicos para me auxiliar no uso do sistema.

1 2 3 4 5

5 - Eu acho que todas funções do sistema estão muito bem desenvolvidas.

1 2 3 4 5

6 - Eu acho que o sistema apresenta muita inconsistência.

1 2 3 4 5

7 - Eu imagino que as pessoas aprenderão como usar esse sistema rapidamente.

1 2 3 4 5

8 - Eu achei o sistema atrapalhado de usar.

1 2 3 4 5

9 - Eu me senti confiante ao usar o sistema.

1 2 3 4 5

10 - Eu precisei aprender várias coisas novas antes de conseguir usar o sistema.

1 2 3 4 5

APÊNDICE B

Questionário de viabilidade do software

Preview questionário

Título: Viabilidade do software

Descrição: Este questionário tem como objetivo avaliar a viabilidade deste software em relação ao sistema atual de ouvidoria da AMF, que é disponibilizado através de uma urna.

1 - Você já teve a intenção de se manifestar à AMF ou de recorrer à Ouvidoria, mas não soube como ou desistiu?

- Sim, tive a intenção, porém desisti Não, nunca tive essa intenção
- Sim, tive a intenção e me manifestei Sim, tive a intenção, porém não soube como me manifestar

2 - Você sabia que a AMF possui um sistema de Ouvidoria? Se sim, você já a utilizou?

- Conheço o sistema e já utilizei Conheço o sistema, porém não utilizei Desconheço o sistema

3 - Qual foi sua satisfação em relação aos serviços prestados pela atual Ouvidoria da faculdade?

- Muito Insatisfeito Insatisfeito Indiferente Satisfeito Muito satisfeito
- Nunca utilizei

4 - Você julga o desenvolvimento deste software de Ouvidoria como algo que irá agregar valor à AMF?

- Não agregará valor Agregará muito valor Agregará pouco valor

5 - Você vê um avanço significativo da AMF em relação ao atendimento ao cliente com este novo sistema?

- Nada significativo Pouco significativo Moderadamente significativo Muito significativo
- Extremamente significativo

6 - Quão bem-sucedido é o software na realização das funções que ele se propõe a fazer se comparado ao sistema de urnas atualmente utilizado pela AMF?

- Nada bem-sucedido Pouco bem-sucedido Moderadamente bem-sucedido
- Muito bem-sucedido Extremamente bem-sucedido

7 - Deixe abaixo alguma recomendação ou comentário referente a esta ideia.

Resposta