



UFOP

Universidade Federal
de Ouro Preto

**Universidade Federal de Ouro Preto
Instituto de Ciências Exatas e Aplicadas
Departamento de Computação e Sistemas**

**Plataforma de Suporte Gamer
Baseada na Comunidade de Usuários**

Ítalo Almeida Aguiar

**TRABALHO DE
CONCLUSÃO DE CURSO**

ORIENTAÇÃO:
Rafael Frederico Alexandre

**Dezembro, 2019
João Monlevade–MG**

Ítalo Almeida Aguiar

**Plataforma de Suporte Gamer Baseada na
Comunidade de Usuários**

Orientador: Rafael Frederico Alexandre

Monografia apresentada ao curso de Engenharia de Computação do Instituto de Ciências Exatas e Aplicadas, da Universidade Federal de Ouro Preto, como requisito parcial para aprovação na Disciplina “Trabalho de Conclusão de Curso II”.

Universidade Federal de Ouro Preto

João Monlevade

Dezembro de 2019

SISBIN - SISTEMA DE BIBLIOTECAS E INFORMAÇÃO

A282p Aguiar, Italo Almeida .
Plataforma de suporte gamer baseada na comunidade de usuários.
[manuscrito] / Italo Almeida Aguiar. - 2019.
53 f.: il.: color., tab..

Orientador: Prof. Dr. Rafael Frederico Alexandre.
Monografia (Bacharelado). Universidade Federal de Ouro Preto.
Instituto de Ciências Exatas e Aplicadas. Graduação em Engenharia de
Computação .

1. Engenharia de software. 2. Jogadores de videogames. 3. Jogos eletrônicos - Serviços de suporte (Gerenciamento). 4. Projeto de produto.
5. Serviços ao cliente. I. Alexandre, Rafael Frederico. II. Universidade Federal de Ouro Preto. III. Título.

CDU 004.41

Bibliotecário(a) Responsável: Flavia Reis - CRB6-2431



MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE OURO PRETO
REITORIA
INSTITUTO DE CIÊNCIAS EXATAS E APLICADAS
DEPARTAMENTO DE COMPUTAÇÃO E SISTEMAS



FOLHA DE APROVAÇÃO

Ítalo Almeida Aguiar

Plataforma de Suporte Gamer Baseada na Comunidade de Usuários

Monografia apresentada ao Curso de Engenharia da Computação da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de Bacharel.

Aprovada em 16 de dezembro de 2019

Membros da banca

Doutor - Rafael Frederico Alexandre - Orientador (Universidade Federal de Ouro Preto)
Mestre - Bruno Cerqueira Hott (Universidade Federal de Ouro Preto)
Doutor - Filipe Nunes Ribeiro - (Universidade Federal de Ouro Preto)
Doutor - George Henrique Godim da Fonseca (Universidade Federal de Ouro Preto)

Rafael Frederico Alexandre, orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 25/03/2021.



Documento assinado eletronicamente por **Rafael Frederico Alexandre, PROFESSOR DE MAGISTERIO SUPERIOR**, em 25/03/2021, às 16:40, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **0151555** e o código CRC **2BF85612**.

Referência: Caso responda este documento, indicar expressamente o Processo nº 23109.002798/2021-54

SEI nº 0151555

R. Diogo de Vasconcelos, 122, - Bairro Pilar Ouro Preto/MG, CEP 35400-000
Telefone: - www.ufop.br

dedicado à minha família

Agradecimentos

Primeiramente agradeço a Deus, por me conceder dádivas incomensuráveis, permitindo-me a concepção não somente desta obra, mas de inúmeras nesta jornada acadêmica.

Agradeço a meus pais, irmão, amigos e familiares por estarem sempre me apoiando e incentivando cada passo neste longo caminho.

Não poderia, em hipótese alguma, deixar de agradecer ao Prof. Dr. Rafael Frederico Alexandre, por todas as nossas discussões e conversas, das quais surgiram grandes ideias. Obrigado por aceitar me direcionar nesta etapa.

Um agradecimento especial a Jeremy VanTassel, pela paciência, prontidão e colaboração ao longo de nossas conversas.

A todos vocês meu sincero obrigado.

“Science is more than a body of knowledge; it is a way of thinking.”

— Carl Sagan (1934 – 1996),
in: The Demon-Haunted World: Science as a Candle in the Dark.

Resumo

Usuários com problemas no uso de certas tecnologias é algo atualmente comum. Isso não é diferente para novos jogos ou plataformas de jogos lançados. Prover um atendimento rápido e eficiente pode ser uma grande ação estratégica para a empresa, e, trazer a própria comunidade para a solução pode ser muito eficaz para solucionar problemas. Este trabalho propõe o desenvolvimento de uma plataforma de suporte voltada para o mercado consumidor de entretenimento eletrônico, onde a comunidade é um pilar fundamental na resolução dos problemas dos clientes. Para isso, a plataforma proposta se baseia nas soluções adotadas por empresas relacionadas a entretenimento eletrônico e o *feedback* reportado pelos seus próprios usuários. Ao final, serão apresentadas as telas e as escolhas de *design* e projeto do produto final.

Abstract

Users with problems using certain technologies is a common issue. This is no different for new games or game platforms released. Providing fast and efficient service can be a great strategic action for the company, and bringing the community itself to the solution can be very effective in solving problems. This paper proposes the development of a support platform focused on the electronic entertainment market, where the community is a fundamental pillar in solving customer problems. For this, the proposed platform is based on the solutions adopted by companies related to electronic entertainment and the feedback reported by their own users. At the end, will be presented the screens and choices of design and design of the final product.

Lista de ilustrações

Figura 1 – Recriação do jogo <i>Tennis for Two</i> pelo Laboratório Nacional de BrookHaven.	16
Figura 2 – Logotipo e <i>slogan</i> do programa <i>XBOX Ambassadors</i>	21
Figura 3 – Anúncio de prêmios feito pelo <i>XBOX Ambassadors</i> nas redes sociais.	23
Figura 4 – Postagem de um usuário no fórum dos embaixadores sugerindo uma forma de centralização das atividades do programa.	24
Figura 5 – Rótulos de reconhecimento nos fóruns da HP.	27
Figura 6 – Rótulos de nível nos fóruns da HP.	28
Figura 7 – Arquitetura do projeto desenvolvido distribuído em camada Cliente e camada Servidor.	31
Figura 8 – Arquitetura da Plataforma Universal do Windows – UWP: Recursos e dispositivos suportados.	31
Figura 9 – Diagrama da estrutura do padrão de projeto MVVM.	36
Figura 10 – Exemplo de utilização do padrão de projeto <i>Command</i> juntamente ao padrão MVVM.	37
Figura 11 – Diagrama da estrutura do padrão de projeto MVC.	38
Figura 12 – Exemplo de código de teste escrito para verificar o comportamento do método <i>GetAsync</i> para um parâmetro inválido.	40
Figura 13 – Tela inicial do aplicativo.	41
Figura 14 – Menu de opções lateral do aplicativo.	42
Figura 15 – Tela do perfil do usuário colaborador.	43
Figura 16 – Tela de Estatísticas do usuário colaborador.	44
Figura 17 – Placares de líderes da comunidade.	45
Figura 18 – Tela de mensagens instantâneas com o usuário.	46
Figura 19 – Tela correspondente ao recurso de Fórum integrado ao aplicativo.	47
Figura 20 – Exemplo de notificação emitida pelo aplicativo para um novo evento.	47
Figura 21 – Ferramenta de busca de códigos de erro e páginas de suporte.	48
Figura 22 – Exemplo de diálogo exibido para o usuário na ocorrência de um erro de rede.	49
Figura 23 – Captura de tela do <i>Hub</i> de comentários dos serviços da loja do <i>Windows e XBOX</i>	49

Lista de tabelas

Tabela 1 – Comparativo dos recursos e funcionalidades das ferramentas de outras empresas e a implementação proposta.	29
--	----

Lista de abreviaturas e siglas

APS	<i>American Physical Society</i>
BNL	<i>Brookhaven National Laboratory</i>
EUA	Estados Unidos da América
US	<i>United States</i>
PS	PlayStation
VR	<i>Virtual Reality</i>
HP	Hewlett Packard
API	<i>Application Programming Interface</i>
SPA	<i>Single Page Application</i>
IoT	<i>Internet of Things</i>
SMF	<i>Simple Machines Forum</i>
SAC	Serviço de atendimento ao consumidor
IDE	<i>Integrated Development Environment</i>
UWP	<i>Universal Windows Platform</i>
MVVM	<i>Model-View-ViewModel</i>
PAX	<i>Penny Arcade Expo</i>
E3	<i>Electronic Entertainment Expo</i>
BLOB	<i>Binary Large Object</i>
SSL	<i>Secure Sockets Layer</i>
HTTP	<i>Hypertext Transfer Protocol</i>
REST	<i>Representational State Transfer</i>
MVC	<i>Model-View-Controller</i>
WPF	<i>Windows Presentation Foundation</i>

XAML	<i>Extensible Application Markup Language</i>
XML	<i>Extensible Markup Language</i>
DLL	<i>Dynamic-link library</i>
IOS	<i>iPhone Operational System</i>
EA	<i>Electronic Arts</i>
XP	<i>Experience Points</i>
HTML	<i>HyperText Markup Language</i>
ASP	<i>Active Server Pages</i>

Lista de símbolos

®	Marca Registrada
™	Trade Mark (Marca Comercial)

Sumário

1	INTRODUÇÃO	16
1.1	Identificação do Problema	17
1.2	Objetivos	18
1.3	Organização do trabalho	19
2	REVISÃO BIBLIOGRÁFICA	20
2.1	<i>Brand Community</i>	20
2.2	<i>Activision Ambassador Program</i>	21
2.3	<i>XBOX Ambassadors™</i>	21
2.4	<i>Answers HQ</i>	24
2.5	A comunidade como parte da solução	25
2.5.1	Engajamento	26
2.5.2	Recompensas	26
2.5.3	Ouvindo a comunidade: <i>feedback</i>	28
2.6	Comparativo entre Ferramentas	29
3	DESENVOLVIMENTO	30
3.1	Definindo a arquitetura do projeto	30
3.2	Escolhendo tecnologias de desenvolvimento	30
3.3	Requisitos do Software	32
3.3.1	Requisitos Funcionais	32
3.3.2	Requisitos não Funcionais	34
3.4	Padrões de Projeto	35
3.4.1	MVVM	35
3.4.2	<i>State</i>	37
3.4.3	<i>Command</i>	37
3.4.4	MVC	37
3.5	Reuso de <i>software</i>	38
3.6	Desenvolvimento do Servidor	39
3.7	Testes de <i>Software</i>	39
3.7.1	Testes de Unidade	40
4	RESULTADOS	41
4.1	<i>Design</i>	41
4.2	Perfil do colaborador	43
4.3	Placares de Líderes	44

4.4	Mensagens Instantâneas	45
4.5	Fórum de Discussão	46
4.6	Sistema de Notificação	47
4.7	Ferramentas de apoio ao suporte	48
4.8	Tratamento de erros na interface gráfica	48
4.9	Coletando o <i>Feedback</i> do Usuário	49
5	CONCLUSÃO	50
5.1	Trabalhos Futuros	50
	REFERÊNCIAS	51

1 Introdução

Lidamos com sistemas e *softwares* todos os dias. Frequentemente estão tão inseridos em nosso dia a dia que mal podemos percebê-los. Ir ao banco e sacar dinheiro em um caixa eletrônico, fazer um telefonema, pagar uma compra no supermercado, utilizar um elevador, ou mesmo assistir televisão. Todas estas tarefas, por mais simples que possam parecer, possuem softwares envolvidos de alguma forma, seja em seu projeto, desenvolvimento, ou mesmo no controle e funcionamento.

Dentre as muitas categorias de *software*, existem aqueles destinados ao entretenimento. Este são conhecidos como jogos digitais. Segundo uma publicação da APS - *American Physical Society* datada de outubro de 2008, o primeiro video-game conhecido foi criado em 1958. Desenvolvido no Laboratório Nacional de Brookhaven, nos EUA, pelo físico William Higinbotham, o jogo chamado de "*Tennis for Two*" consistia em uma versão do tênis de quadra (OUELLETTE, 2008). A Figura 1 mostra uma recriação do jogo original de 1958 em comemoração ao aniversário de 50 anos de sua criação (BNL, 2013).



Figura 1 – Recriação do jogo *Tennis for Two* pelo Laboratório Nacional de BrookHaven.

Fonte: BNL.

De 1958 até os tempos atuais houveram grandes evoluções tecnológicas em todas as áreas da ciência e tecnologia. Os jogos atuais possuem um nível de complexidade e tecnologia consideravelmente maiores que seus predecessores. Porém, é fato que quanto maior a complexidade de um sistema, mais susceptível a falhas ele pode ser (DELAMARO; JINO; MALDONADO, 2013), e não é incomum encontrar usuários apresentando problemas relacionados a jogos ou console de jogos em alguns canais da Internet, como fóruns e redes sociais.

Como mencionado anteriormente, a Internet possui alguns canais onde os usuários podem expressar seus problemas. Alguns destes canais são próprios para que os usuários com problemas possam obter ajuda especializada. Podemos classificar esses canais em duas vertentes diferentes. De um lado estão os canais oficiais, disponibilizados pelas empresas desenvolvedoras do jogo, plataforma ou tecnologia. Do outro lado estão os canais não oficiais, que não estão diretamente vinculados aos desenvolvedores e fabricantes, mas encorajam a solução de problemas por outros usuários.

As plataformas não oficiais são encontradas geralmente na forma de fóruns de discussão, *blogs* e video-tutoriais. As soluções provenientes destas plataformas são criadas por usuários da mesma tecnologia ou usuários com afinidade pela mesma. Embora as soluções provenientes destas plataformas não sejam capazes de solucionar todos os problemas dos usuários, para os mais comuns, a solução poderá ser tão eficiente quanto a solução de um canal oficial. Por outro lado, plataformas oficiais podem oferecer um nível de suporte especializado capaz de promover a resolução dos problemas mais complexos enfrentados pelos usuários. É composto por uma equipe de pessoas treinadas pela empresa para fornecer soluções definitivas aos problemas apresentados. O suporte especializado pode utilizar tecnologias como telefone, *e-mail*, mensagens instantâneas, fóruns de discussão, redes sociais e até mesmo ferramentas mais avançadas como acesso remoto ao dispositivo do usuário.

1.1 Identificação do Problema

Do ponto de vista da empresa dona da tecnologia, prover soluções para os problemas de seus usuários requer profissionais habilitados para isso. Portanto é necessário um treinamento especializado para cada um dos membros que integrarão a equipe de suporte (MINCER, 1962). Toda essa equipe e treinamento traz para a empresa uma série de custos, mesmo após o treinamento. Em muitos casos, o tamanho da equipe pode ser insuficiente para a demanda por suporte, o que pode exigir que a empresa tenha mais gastos na contratação de mais profissionais e seu respectivo treinamento.

A satisfação do usuário é algo fundamental para manter sua adesão ao produto ou plataforma. O problema reside em quão rápido um problema pode ser solucionado, e principalmente o quão eficaz é a solução. Além disso, o usuário deve sentir que há uma relação de respeito e atenção ao ser atendido perante o seu problema.

Desta forma, a existência de plataformas de suporte online torna-se, um recurso estratégico para a empresa. Quando uma plataforma de suporte se integra aos usuários de seus produtos, a comunidade passa a fazer parte da resolução dos problemas que eventuais usuários podem encontrar. Isso permite uma redução de custos para a empresa, onde geralmente, problemas em mais baixo nível de complexidade, podem não exigir a

intervenção de mão de obra especializada do suporte.

Segundo uma pesquisa de 2019 realizada pelo site Reclame Aqui¹, foram registradas 218.715 queixas de usuários não atendidos pelas empresas no período de janeiro a outubro de 2019. Segundo o site, esse número reflete um aumento de 24,3% em relação ao mesmo período de 2018 (RECLAMEAQUI, 2019). Isso torna ainda mais evidente a necessidade de mecanismos eficientes de atendimento ao usuário, como plataformas online especializadas nesta tarefa.

1.2 Objetivos

Este trabalho tem por objetivo propor o desenvolvimento de uma plataforma de suporte destinada ao público consumidor de entretenimento eletrônico. A plataforma será constituída de um conjunto de subsistemas diferentes capazes de fornecer vários canais de suporte para o usuário final, sendo completamente centralizado para a equipe de suporte e voluntários da comunidade. Isso significa que para as pessoas responsáveis por atender a usuários com problemas, todos os subsistemas estarão integrados em um único painel de controle, mesmo que os canais de suporte estejam distribuídos em plataformas ou sistemas distintos.

A plataforma proposta possui um maior foco em trazer usuários mais experientes como parceiros e colaboradores, capazes de contribuir positivamente na experiência de outros usuários. Os principais pontos deste trabalho são:

- Desenvolver uma plataforma de suporte multi dispositivo, onde os usuários poderão ser capazes de colaborar na solução de problemas, através de computadores, *tablets*, *smartphones*, etc. Permitindo mobilidade entre dispositivos, usuários poderão enviar suas colaborações de onde estiverem através de seus *smartphones*.
- Incorporar diferentes canais de suporte, como fóruns, mensagens instantâneas e redes sociais. Isso permite que os usuários tenham o direito de escolher qual é a melhor forma de se comunicar, evitando a saturação de clientes em um único canal e permitindo manter as expectativas do usuário dentro do esperado.
- Modelar o desenvolvimento pensando nos usuários como parte da solução de problemas. Permitir que a comunidade traga para a plataforma suas experiências e compartilhe suas opiniões, ajudando a todos que possam estar com problemas técnicos.

¹ <http://www.reclameaqui.com.br>

1.3 Organização do trabalho

No **Capítulo 2** deste trabalho serão abordadas algumas das plataformas de suporte que mais se destacaram por trazer a comunidade usuária de um produto ou serviço como parte da resolução de problemas. No **Capítulo 3** serão introduzidos alguns conceitos iniciais que irão reger o desenvolvimento. Adiante serão abordadas as tecnologias e etapas de engenharia de *software* empregados na construção da solução aqui proposta. O **Capítulo 4** é o responsável por apresentar a solução desenvolvida neste trabalho. Isso inclui o *design* e *layout* adotado na concepção das interfaces gráficas de usuário. Este capítulo também justifica a arquitetura distribuída em cliente e servidor, mostrando o funcionamento adequado de cada uma das telas que compõe o *software* final. Os **Capítulos 5** e **6** apresentam as conclusões para o *software* proposto e os possíveis trabalhos futuros que poderão ser realizados a tomar este trabalho como ponto de partida.

2 Revisão bibliográfica

Sempre que um novo jogo eletrônico é lançado no mercado, a empresa responsável disponibiliza canais de suporte para atender a seus usuários. Este é inclusive, um requisito obrigatório para a publicação em algumas plataformas de distribuição, como por exemplo, a *Windows Store* (MICROSOFT CORPORATION, 2018).

Geralmente associa-se a responsabilidade pelo suporte exclusivamente aos desenvolvedores do produto, no caso, o jogo. Porém, quando publicado através de uma plataforma de distribuição, é comum que a plataforma também possua um canal de suporte com usuários capazes de ajudar na solução de problemas. Este é o caso da loja virtual de jogos *Steam*¹, por exemplo.

Existem no mercado muitas empresas que possuem suas próprias plataformas de suporte, porém, a grande maioria é voltada exclusivamente ao atendimento entre cliente e funcionário. Algumas empresas podem até possuir plataformas de fórum online onde sua comunidade pode expressar opiniões e ajudar outros usuários. Infelizmente, empresas que de fato reconhecem as contribuições extraordinárias de usuários da comunidade são relativamente raras.

2.1 *Brand Community*

Brand Community, inicialmente introduzido por Muniz e O'Guinn, é um termo utilizado para descrever toda uma comunidade de usuários formada ao redor de uma marca, produto, empresa, ou tecnologia (MCALEXANDER; SCHOUTEN; KOENIG, 2002). Este tipo de comunidade não está geograficamente vinculada e se baseia nas relações sociais acerca dos gostos pessoais dos consumidores (MUNIZ; O'GUINN, 2001).

As *Brand Communities*, assim como outras comunidades, são marcadas por possuírem consciência compartilhada, rituais e tradições, além de um senso de responsabilidade moral (MUNIZ; O'GUINN, 2001). Para uma empresa, a comunidade construída acerca de sua marca é a maior rede social, capaz de desempenhar um papel fundamental no legado da marca (MUNIZ; O'GUINN, 2001).

Brand Communities são capazes de permitir a execução de testes de mercado ao reunir diferentes níveis de usuários familiarizados com a marca (AMINE; SITZ, 2004). Além disso, *Brand Communities* são importantes fontes de difusão de informação, sendo um recurso precioso do *Marketing* da empresa (MCALEXANDER; SCHOUTEN; KOENIG, 2002).

¹ <https://help.steampowered.com/>

2.2 Activision Ambassador Program

A Activision², uma empresa desenvolvedora de jogos digitais, famosa por franquias como *Call of Duty*³ possui um programa de suporte comunitário denominado *Activision Ambassador Program*.

A ideia por trás do programa é permitir que membros e jogadores dedicados possam tornar a comunidade um lugar melhor. Para isso, os embaixadores da Activision ajudam os outros jogadores a solucionar problemas, fornecer conselhos, dicas e truques. Entre os benefícios do programa estão o acesso a conteúdo exclusivo, direito de participar de eventos e desafios especiais com premiações como equipamentos e camisetas (ACTIVISION, 2019).

Para se tornar um embaixador da Activision, o jogador pode fazer o pedido de inscrição através da página dos embaixadores no site da empresa. A empresa realizará uma rigorosa análise de candidatos, podendo chegar a uma espera de até 4 semanas para a conclusão do processo. Atualmente o programa é suportado somente no idioma inglês e está disponível apenas no Estados Unidos, Canadá, Reino Unido e Austrália (ACTIVISION, 2019).

2.3 XBOX Ambassadors™



Figura 2 – Logotipo e *slogan* do programa *XBOX Ambassadors*.

Fonte: XBOX WIRE

Lançado em meados de 2011, *XBOX Ambassadors* é um programa da Microsoft para sua plataforma de jogos *XBOX*. O lema do programa é “*Lets make game fun for everyone*”, em tradução livre, “Vamos fazer os jogos divertidos para todos”. O logotipo e o *slogan* do programa podem ser observados na Figura 2. O programa permite que usuários da plataforma recebam um título de distinção, denominado “*XBOX Ambassador*”,

² <https://www.activision.com/>

³ <https://www.callofduty.com/>

dentro da comunidade *XBOX Live* e o fórum de suporte da Microsoft, o *Microsoft Answers* (MICROSOFT CORPORATION, 2011).

Qualquer usuário da plataforma pode se tornar um embaixador. Para isso é necessário atender a 3 requisitos (MICROSOFT CORPORATION, 2011):

- Ser maior de 17 anos.
- Possuir no mínimo 1500 pontos no perfil do *XBOX*. Estes pontos são conquistados ao se jogar qualquer jogo na plataforma.
- Não possuir nenhum tipo de ação ligada a má conduta dentro da comunidade no último ano.

Um embaixador deve ser um usuário exemplar dentro da comunidade. Deve ser capaz de tratar respeitosamente todos com quem se relaciona. Os embaixadores estão sujeitos ao código de conduta determinado pelo programa, com pena de banimento em caso de descumprimento das regras nele determinado (MICROSOFT CORPORATION, 2019b).

Os embaixadores compartilham seus conhecimentos e experiências dentro da plataforma *XBOX* com outros usuários. Além disso, os embaixadores têm por objetivo tentar ajudar a todos que o procurarem e também participar de atividades que tornam a plataforma um lugar seguro, inclusivo e divertido para todos. Existem várias atividades dentro do programa que contribuem para uma comunidade saudável, um exemplo é manter uma presença ativa nos fóruns de suporte do *XBOX* ajudando outros usuários. Embaixadores podem também criar conteúdo, como vídeos tutoriais que possam contribuir positivamente na solução de problemas (MICROSOFT CORPORATION, 2019b).

A plataforma dos embaixadores possui um sistema de *ranking* com níveis divididos em bronze, prata, ouro, platina, diamante e um nível adicional para os 100 primeiros do *ranking*, o *top 100*. Os embaixadores recebem pontos de experiência por completar atividades e missões dentro da plataforma. Esses pontos são utilizados no *ranking* de embaixadores.

As atividades do programa consistem em tarefas como responder a questões nos fóruns ou no Twitter; responder a *quizzes* sobre o programa, a comunidade ou sobre a plataforma *XBOX*; submeter e votar em vídeos pertinentes ao programa; conversar com outros usuários da comunidade, etc. O programa também conta com missões, que nada mais são que um agrupamento de atividades que fornecem pontos de experiência ao serem completadas. Existem missões diárias, semanais e mensais (MICROSOFT CORPORATION, 2019b). Isso pode ser visto como uma forma de incentivar o embaixador a manter-se ativo no programa todos os dias.

É comum que os usuários recebam algum tipo de recompensa ao avançarem nos níveis do ranking do programa. Para os níveis mais baixos, o programa geralmente premia os embaixadores com recompensas digitais, como itens de avatar, que podem ser utilizados no perfil público da plataforma *XBOX*, a *XBOX Live*. Níveis mais altos podem contar com recompensas físicas que são posteriormente enviadas para o endereço do embaixador. É comum que o programa divulgue em suas redes sociais oficiais algumas das premiações que estão ocorrendo no momento, como pode ser visto na Figura 3.



Figura 3 – Anúncio de prêmios feito pelo *XBOX Ambassadors* nas redes sociais.

Fonte: Microsoft XBOX Ambassadors

O programa também conta com uma plataforma de sorteio de itens, físicos ou digitais, que todos os embaixadores podem participar. Alguns tipos especiais de missões do programa recompensam o usuário com *tickets* de sorteio ao invés de pontos de experiência. Esses *tickets* de sorteio podem ser depositados em itens à escolha do embaixador por um breve período anterior ao sorteio (MICROSOFT CORPORATION, 2019b). Ao ganhar um sorteio a recompensa é enviada para o endereço do usuário, no caso de recompensas físicas. Para recompensas digitais, um código é enviado ao *e-mail* do embaixador para que ele possa resgatar o item na loja virtual do *XBOX*.

O programa também está presente fisicamente através dos principais eventos da comunidade de jogos, como a série de conferências *PAX*⁴, *E3*⁵, *XBOX FanFest*, etc. Estes eventos são oportunidades para que os embaixadores possam se conhecer pessoalmente além de conhecer as pessoas que gerenciam o programa, chamados de *Community Managers*, ou gerentes de comunidade. Cada um destes eventos conta com um "Stand" dos embaixadores, onde há espaços para conversas, fotos, estações de jogos, podendo até mesmo ter roletas de sorteio de prêmios para os embaixadores presentes nos eventos (MICROSOFT CORPORATION, 2019e).

⁴ <https://www.paxsite.com/>

⁵ <https://www.e3expo.com/>

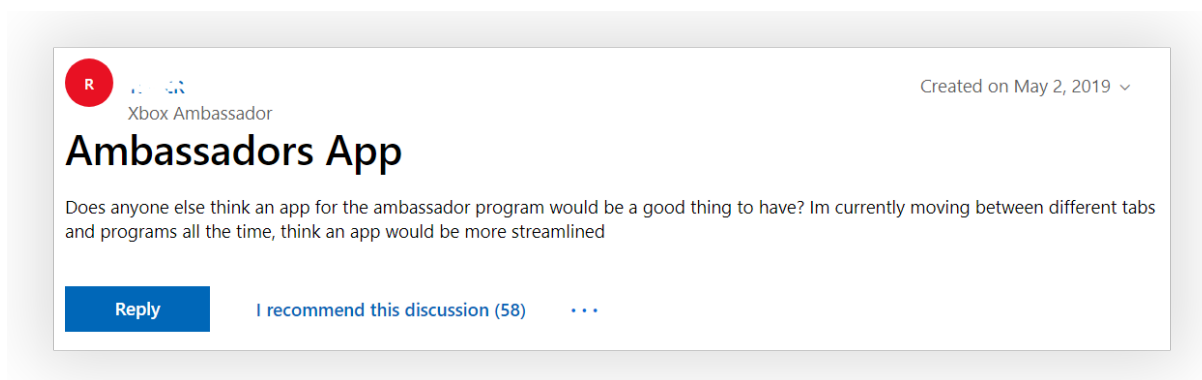


Figura 4 – Postagem de um usuário no fórum dos embaixadores sugerindo uma forma de centralização das atividades do programa.

Fonte: Microsoft XBOX Ambassadors.

Embora a Microsoft tenha obtido muito sucesso com o programa *XBOX Ambassadors*, atualmente com mais de 100.000 participantes, existem algumas questões levantadas pelos próprios embaixadores com relação ao programa. Uma destas questões critica a forma descentralizada com que as atividades do programa são realizadas. Isso se deve ao fato de que as plataformas do programa estão espalhadas por diferentes sítios, exigindo um número maior de abas abertas no navegador. Um exemplo deste tipo de *feedback* pode ser observado na Figura 4 retirada de uma postagem no fórum dos embaixadores. O nome do autor da postagem foi apagado para preservar sua identidade (MICROSOFT CORPORATION, 2019a).

Uma outra questão é que os embaixadores não recebem notificações quando novas atividades e missões estão disponíveis. Isso faz com que os embaixadores tenham que constantemente atualizar as páginas da plataforma em busca de mudanças.

2.4 *Answers HQ*

Answers HQ é um fórum de suporte criado pela Electronic Arts, ou simplesmente EA, uma empresa desenvolvedora de jogos digitais. A ideia por trás do fórum é encorajar usuários a ajudarem outros membros da comunidade com problemas em jogos desenvolvidos pela empresa (ELECTRONIC ARTS, 2019).

O fórum utiliza um sistema de pontos de experiência, XP. Os pontos de experiência são acumulados pelo usuário ao realizar atividades no fórum, como criar postagens, responder a usuários, receber uma resposta útil, etc (ELECTRONIC ARTS, 2019).

Os pontos de experiência permitem que o usuário possa escalar os níveis definidos pelo fórum, sendo eles, *Newbie*, *Novice*, *Apprentice*, *Guide*, *Pro*, *Expert*, *Master* e *Guru*. Além dos níveis, o programa também concede insígnias para o usuário ao completar algumas atividades específicas (ELECTRONIC ARTS, 2019).

Deferentemente dos programas *Activision Ambassadors* e *XBOX Ambassadors*, o *Answers HQ* não promove nenhum tipo de premiação com artefatos físicos ou digitais (ELECTRONIC ARTS, 2019).

2.5 A comunidade como parte da solução

Aproveitar o conhecimento e experiência dos próprios usuários da tecnologia é uma estratégia muito útil quando deseja-se prestar um bom atendimento. É comum que usuários mais ávidos à tecnologia sejam capazes de solucionar os próprios problemas e dificuldades ao utilizar determinado produto ou tecnologia. Este tipo de usuário pode ser uma peça chave quando são criadas oportunidades para se compartilhar experiências e conhecimentos.

Existem várias naturezas para um problema dentro de uma plataforma. Há casos em que o problema não pode ser resolvido a nível do usuário, como por exemplo, um *bug* no *software*. Neste caso, a solução do problema depende unicamente do desenvolvedor do produto, que deverá efetuar as correções e disponibilizar uma atualização para seus usuários (DELAMARO; JINO; MALDONADO, 2013). Há também problemas em um nível de suporte especializado, exigindo que um profissional da empresa dona do produto ou serviço seja acionado. Este é o caso de problemas relacionados a transações financeiras, como reembolsos, por exemplo.

Problemas de natureza física também podem se apresentar para os usuários, porém, esta classe só está relacionada a produtos físicos, como computadores, celulares, *tablets*, videogames, etc. Este nível de problema também exige suporte especializado para o produto. A diferença é que o problema, geralmente, não pode ser sanado na residência do usuário, ou local de instalação do produto. O produto físico avariado deverá ser transportado para uma assistência técnica onde os reparos serão feitos. Uma exceção a esta regra são as chamadas visitas domiciliares, em que um técnico especializado atende a um chamado de reparo e visita a localização do produto para efetuar os reparos no local.

Os problemas a nível de usuário geralmente podem ser solucionados através de configurações no produto ou executando uma série de procedimentos conhecidos. Quando um usuário se depara com outro membro da comunidade enfrentando alguma dificuldade anteriormente já solucionada, o compartilhamento da solução e procedimentos realizados na situação podem prover uma resposta eficiente e rápida.

Não é incomum que os usuários mais experientes divulguem seus conhecimentos através de certos canais na Internet, principalmente relacionado a *softwares*. Existem vários blogs de tecnologia que escrevem artigos detalhando passo a passo a solução dos problemas mais comuns de determinada plataforma. O fórum *XDA Developers*⁶, por exemplo, é

⁶ <https://www.xda-developers.com/>

uma referência em dispositivos móveis, disponibilizando inclusive, modificações da própria comunidade para determinados dispositivos.

Em alguns casos, vídeo tutoriais também são disponibilizados explicando a solução detalhada para o problema. As redes sociais também apresentam comunidades de usuários onde os problemas podem ser discutidos de forma coletiva.

Embora a empresa desenvolvedora disponibilize canais oficiais de suporte. Geralmente estes canais estão sujeitos a uma equipe de suporte limitada em números. Em uma situação em que um elevado número de usuários com problemas contacta o suporte, possivelmente uma parcela destes usuários poderá enfrentar longas esperas em filas para serem atendidos. Para contornar este problema, os canais de suporte podem adotar uma estratégia de convidar outros usuários a fornecerem soluções.

Plataformas online de suporte, como fóruns de discussão já são conhecidos por permitir que qualquer pessoa entre em uma conversa e envie seus comentários. Ao mesmo tempo, o autor da conversa, portador original do problema, pode marca-lo como solucionado caso alguma das respostas tenha atendido ao esperado. Isso permite reduzir consideravelmente a espera por suporte e possivelmente prover uma solução rápida. Para os casos não solucionados o suporte especializado presente dentro da plataforma pode assumir a discussão para resolver por definitivo o problema.

2.5.1 Engajamento

Manter uma comunidade de suporte ativa composta pelos próprios usuários é um ponto que deve ser observado com atenção. Usuários que compõe a comunidade de suporte podem deixá-la a qualquer momento, principalmente se não houver algum tipo de reconhecimento acerca de suas contribuições (REILLIER; REILLIER, 2017).

Em alguns casos, um simples agradecimento oficial ao usuário pode promover um efeito motivador que encoraje o usuário a se dedicar ainda mais e melhorar seu impacto dentro da comunidade. Note que não é necessário nenhum tipo de recompensa financeira ou física para motivar um usuário, embora este também possa ser um meio eficiente de motivação.

A própria plataforma de suporte também pode prover um efeito motivador. O uso de títulos diferenciados, como “parceiro” na plataforma, traz consigo um efeito de reconhecimento natural.

2.5.2 Recompensas

Recompensar o usuário por suas contribuições é uma forma eficiente de motivar a permanência na comunidade de suporte. Existem várias estratégias que determinada empresa pode adotar em seus canais de suporte. Existem estratégias de baixo e alto custo,

e, a adoção de uma não necessariamente exclui a possibilidade de adoção da outra. Para ambos os casos, a empresa poderá realizar um tipo de teste conhecido como Teste A/B na engenharia de *software* (DELAMARO; JINO; MALDONADO, 2013). O Teste A/B consiste em executar e comparar diferentes estratégias de negócio em amostras distinta de usuários. Este tipo de teste permitiria que a empresa analisasse qual é o melhor caminho a ser seguido com base no resultado obtido no teste.

Podemos destacar algumas das principais formas de recompensação de usuários acerca de suas contribuições:

- **Recompensas digitais (Distintivos):** Diferenciar os usuários que compõem a comunidade de suporte é uma parte fundamental na permanência dos usuários da comunidade. O suporte pode adotar uma estratégia de experiência mínima para reconhecer o usuário como “parceiro”. Ao atingir o nível de contribuição mínimo, o usuário pode receber um distintivo em seu perfil. Isso permite a qualquer um diferenciar o usuário dentro da plataforma de suporte.

A Hp promove em seu fórum⁷ de suporte um reconhecimento para os usuários voluntários que mais se destacam por suas contribuições. Para estes é concedido um rótulo distintivo que passa a ser exibido abaixo de cada uma das postagens deste usuário, conforme pode ser observado na Figura 5. Isso garante uma maior confiabilidade em relação à resposta por parte do usuário com problema.

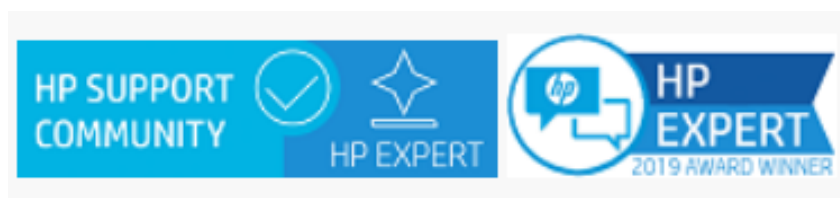


Figura 5 – Rótulos de reconhecimento nos fóruns da HP.

Fonte: Fórum HP.

- **Recompensas de Status (Ranking/Níveis):** Esta é uma estratégia relativamente de baixo custo onde a plataforma de suporte cria um sistema de pontuação e níveis. À medida que o usuário faz contribuições dentro da plataforma, pontos são atribuídos a seu perfil permitindo avançar entre estes níveis. Os níveis podem ser meramente numéricos ou podem atribuir nomenclaturas que refletem a experiência do usuário contribuidor.

Os fóruns de suporte da HP também contam com um sistema de nível de usuários. É possível observar na Figura 6 o rótulo “*Expert*” ao lado do nome do usuário, e um gráfico composto por barras coloridas, estrelas, e o nível numérico indicando o grau

⁷ <https://h30487.www3.hp.com/>

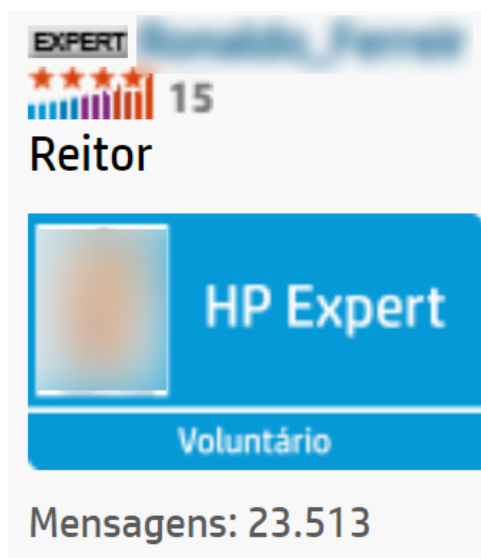


Figura 6 – Rótulos de nível nos fóruns da HP.

Fonte: Fórum HP.

de dedicação do baseado na quantificação das contribuições realizadas. Também é possível observar o rótulo "*Reitor*" como forma de diferenciação entre os demais usuários. Todo esse processo pode ser realizado de forma automática pelo sistema.

- **Recompensas físicas:** Fornecer aos usuários recompensas físicas, como troféus, brindes e artefatos, embora possa ser uma estratégia aparentemente, de custo mais elevado, os resultados podem mostrar um retorno considerável sobre o investimento. Esta talvez seja a forma de reconhecimento mais notável pelo usuário, sendo para a empresa, talvez a ferramenta de motivação mais efetiva.

É importante perceber que, além de motivar o usuário contemplado a continuar com o bom trabalho, esta estratégia também promove uma competição saudável entre os demais participantes da comunidade de suporte, visando também serem reconhecidos.

2.5.3 Ouvindo a comunidade: *feedback*

A empresa deverá ter uma atenção especial com sua comunidade de suporte. É importante disponibilizar um canal onde cada um dos membros da comunidade de suporte sejam encorajados a expressar suas ideias. As opiniões, sugestões e preocupações relatadas devem ser ouvidas, e principalmente, o usuário deve sentir que está sendo ouvido.

A empresa deverá avaliar a possibilidade de implementação de melhorias com base nas requisições recebidas. Nos casos em que uma requisição não possa ser atendida, uma resposta oficial satisfatória por parte da empresa deve ser compartilhada com o usuário,

apontando os motivos para o qual não possa ser implementada (REILLIER; REILLIER, 2017).

2.6 Comparativo entre Ferramentas

Baseando-se nas plataformas proprietárias anteriormente descritas, a Tabela 1 ilustra as diferenças entre estas plataformas e o software aqui proposto. A Tabela 1 é composta de um comparativo entre as funcionalidades e recursos presentes ou ausentes nestas plataformas.

Programa	Recursos				Ranking		Recompensas		Dispositivos			
	Fórum	Chat	Social	E-mail	Pontos	Níveis	Digitais	Físicas	Web	Desktop	Mobile	Console
XBOX Ambassadors	✓	✗	✓	✗	✓	✓	✓	✓	✓	✗	✗	✗
Activision Ambassadors	✓	✓	✗	✗	✓	?	✓	✓	✓	✗	✗	✗
EA Answers HQ	✓	✗	✗	✗	✓	✓	✗	✗	✓	✗	✗	✗
Software Proposto	✓	✓	✓	🕒	✓	✓	✓	✓	✓	✓	🕒	✓

Legenda: ✓ Recurso Presente ✗ Recurso Ausente 🕒 Recurso Agendado ? Dado indisponível

Tabela 1 – Comparativo dos recursos e funcionalidades das ferramentas de outras empresas e a implementação proposta.

No quesito Dispositivos, presente na Tabela 1, cabe ressaltar que todas as plataformas avaliadas são capazes de serem executadas em todos os dispositivos citados na tabela, porém através do navegador de internet. A caracterização como recurso ausente significa que não há um software nativo desenvolvido especificamente para o dispositivo.

Em relação ao programa *XBOX Ambassadors*, no quesito *Chat*, em determinado momento no passado o recurso era suportado pelo programa, porém, atualmente esse recurso não está mais disponível.

As implementações agendadas para o *software* aqui proposto, como exibido na Tabela 1, são baseadas no princípio de reuso de *software*, conforme será abordado mais adiante no **Capítulo 3**.

3 Desenvolvimento

A proposta deste trabalho é o desenvolvimento de uma plataforma de suporte voltada para o público consumidor de jogos eletrônicos. Porém, esta plataforma possui uma maior atenção em trazer a própria comunidade para a solução. Para isso, este trabalho toma como base os pontos positivos que plataformas atuais, como o *XBOX Ambassadors* possuem, tentando contornar os pontos negativos levantados por seus usuários. Nas seções a seguir serão definidos alguns conceitos dos quais serão pilares fundamentais da concepção do *software* final.

3.1 Definindo a arquitetura do projeto

O projeto desenvolvido foi dividido em duas partes, uma do lado do servidor, e a outra, os clientes responsáveis por consumir os dados do servidor.

O servidor é onde toda a lógica de negócio da aplicação será concentrada. Cada uma das aplicações como fórum, *chat*, *e-mail*, é desenvolvida em uma aplicação independente. Isso permite escalar mais facilmente os recursos de computação em nuvem à medida que o número de requisições varia no tempo. Para que isso seja possível, todas as aplicações são desenvolvidas para trabalhar sobre um único servidor banco de dados, capaz de lidar com um alto número de conexões em paralelo. Utilizar o mesmo servidor de banco de dados para todas as aplicações permite compartilhar dados sem que seja necessário estabelecer conexões entre as próprias aplicações. Para que os clientes possam consumir dados do servidor, cada uma das aplicações deve fornecer pontos de acesso para uma *API restful* de acesso aos dados e funcionalidades necessários (MICROSOFT CORPORATION, 2019c).

As APIs de dados disponibilizadas por cada uma das aplicações permitem que diferentes tipos de clientes sejam desenvolvidos. Isso abre possibilidade para o desenvolvimento de aplicações para dispositivos móveis, como *tablets* e *smartphones*; aplicativos do navegador de Internet, como SPAs (aplicações de página única) com *javascript* (MONTEIRO, 2014); *Smart TVs*; videogames, dispositivos *IoT* (Internet das coisas); entre outros. A Figura 7 ilustra a arquitetura adotada para as aplicações cliente e servidor.

3.2 Escolhendo tecnologias de desenvolvimento

Por questões de maior afinidade e conhecimentos prévios, a linguagem de programação C# (*C Sharp*) foi adotada como linguagem principal de desenvolvimento. C# é uma linguagem de programação orientada a objetos e multi paradigma criada pela

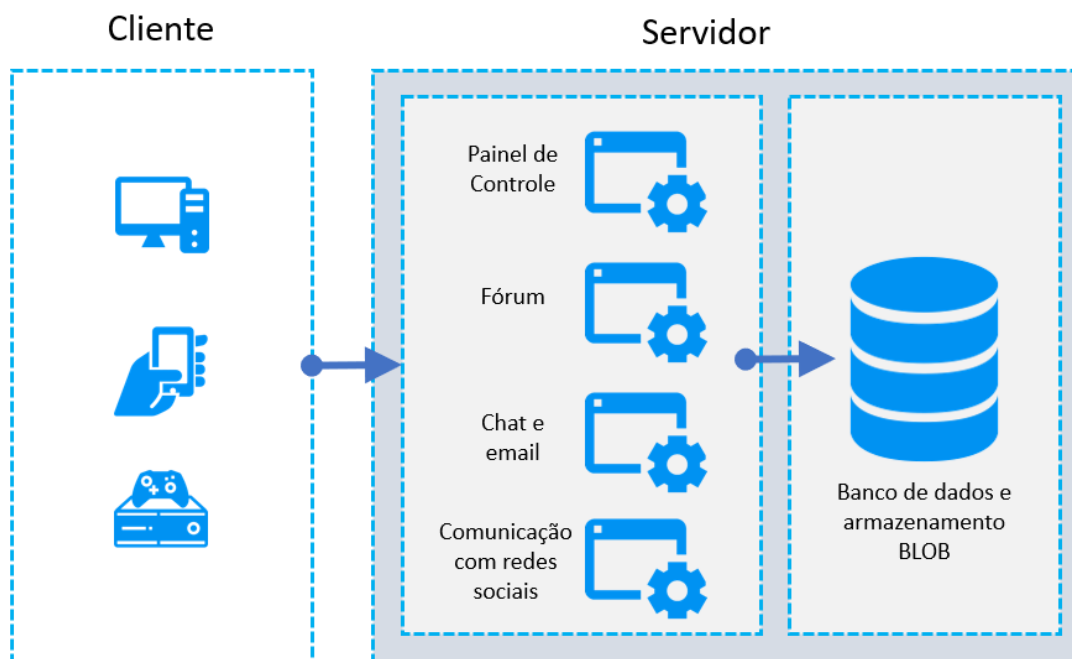


Figura 7 – Arquitetura do projeto desenvolvido distribuído em camada Cliente e camada Servidor.

Microsoft Corporation. Sintaticamente, C# apresenta grande similaridade a linguagens C/C++ e principalmente Java (ECMA INTERNATIONAL, 2017). C# pode ser utilizado no desenvolvimento de vários tipos de *software*, entre eles, aplicativos *desktop*, serviços do *Windows*, aplicativos para dispositivos móveis, *websites* e *webservices*, jogos com *DirectX* ou *OpenGL*, IoT (Internet das coisas) e aplicativos do *XBOX*.



Figura 8 – Arquitetura da Plataforma Universal do Windows – UWP: Recursos e dispositivos suportados.

Fonte: Microsoft.

Dada a escolha da linguagem de programação, no lado do cliente, a escolha da plataforma de publicação foi optar pela Plataforma Universal do *Windows* –UWP. Esta escolha deve-se ao fato de que os Aplicativos Universais do *Windows* permitem que o

mesmo código seja capaz de ser executado em uma gama muito grande de dispositivos (MICROSOFT CORPORATION, 2019g). A Figura 8 ilustra as tecnologias e os dispositivos que a plataforma é capaz de atender.

O ambiente de desenvolvimento integrado, IDE, para este projeto, foi o *Microsoft Visual Studio*. Esta é uma escolha natural dado que o *Visual Studio* é a ferramenta oficial da Microsoft para o desenvolvimento de aplicativos da plataforma UWP (MICROSOFT CORPORATION, 2019d). Atualmente existem três versões do *Visual Studio*, *Enterprise*, *Professional* e *Community*. As duas primeiras versões, *Enterprise* e *Professional*, são versões pagas destinadas a grandes e médias empresas e times de desenvolvimento, respectivamente. A terceira versão, *Community*, é a versão gratuita e livre para a comunidade (MICROSOFT CORPORATION, 2019f). A versão *Community* foi a versão utilizada no desenvolvimento do *software* desenvolvido.

3.3 Requisitos do Software

Requisitos de *software* descrevem as funcionalidades das quais o *software* em desenvolvimento deverá possuir. Normalmente estão associados às necessidades do usuário para realizar determinada tarefa ao qual o *software* se propõe a resolver (SOMMERVILLE, 2007). Os requisitos de *software* podem ser divididos em duas categorias principais: Requisitos Funcionais e Requisitos não Funcionais.

3.3.1 Requisitos Funcionais

Os requisitos funcionais descrevem de forma explícita as funcionalidades que o sistema deve ser capaz de realizar, como por exemplo, edição de textos, reprodução de áudio, leitura de arquivos, etc. Em outras palavras, os requisitos funcionais são as funcionalidades das quais o usuário da aplicação consegue lidar diretamente (SOMMERVILLE, 2007). A seguir são descritos os requisitos de *software* para o projeto.

- **Fóruns de Discussão:** Os fóruns são talvez a principal plataforma de suporte com apoio à comunidade. Geralmente permite que qualquer usuário registrado inicie uma discussão, onde todos podem expressar suas opiniões e experiências sobre o tema. Fóruns podem ser utilizados para discutir qualquer tipo de assunto. Existem fóruns destinados a críticas de cinema, aeromodelos, hardware e tecnologia, automóveis, etc. Em todos os casos, os fóruns, independentemente do tema, podem ser utilizados para resolução de problemas acerca de seu tema central.

Existem plataformas gratuitas de código aberto, como o SMF – *Simple Machines Forum*¹, que podem facilmente ser adotadas por qualquer empresa. O SMF, assim

¹ <https://www.simplemachines.org/>

como várias plataformas de fórum já possui um sistema de distintivo que diferencia usuários comuns de usuários especiais. O SMF também possui um sistema automático de ranking na plataforma. Isso permite que o usuário possa avançar de nível conforme conclui atividades dentro da plataforma. Os níveis além de terem um efeito de reconhecimento também promovem uma distinção do usuário dentro da plataforma. Embora utilizar uma plataforma já existente seja atrativo, o custo das adaptações para necessidades específicas de suporte podem tornar seu uso inviável. Neste trabalho é proposto o uso de uma plataforma de fórum própria.

- **Mensagens instantâneas:** Popularmente conhecido como *chat*, as mensagens instantâneas são um tipo de plataforma de comunicação de tempo real através de mensagens de texto. Em alguns casos, os *chats* podem possuir recursos avançados como envio de imagens e capturas de tela, mensagens de voz e até mesmo vídeo chamadas. As mensagens instantâneas de um chat podem ser um substituto muito eficiente para as tradicionais ligações telefônicas para os números de serviço de atendimento ao consumidor (SAC).

Em alguns casos o *software* do *chat* é equipado com um algoritmo de inteligência artificial capaz de realizar um atendimento prévio instantâneo. A inteligência artificial faz uma análise de linguagem natural sobre as mensagens enviadas pelo usuário e procura tentar fornecer a melhor resposta encontrada no banco de dados. A inteligência artificial, geralmente dotada de aprendizado de máquina, tende a melhorar sua resposta com o que consegue aprender a cada interação humana. Nos casos em que a inteligência artificial é incapaz de fornecer uma resposta aceitável, a conversa do *chat* pode ser transferida para um atendimento humano.

Antes de escalar o atendimento para o suporte especializado, um nível intermediário, destinado à comunidade pode ser criado. Cada usuário da comunidade pode ter acesso às conversas anteriores e decidir se possui o conhecimento adequado para sugerir uma solução ao aceitar participar da conversa de *chat*. Neste caso, um recurso de limite de tempo é importante para que caso nenhum usuário da comunidade aceite a requisição, esta seja imediatamente escalada para o nível seguinte, no caso, o suporte especializado. Esta medida tende a garantir que o usuário não passe por uma longa espera para ser atendido.

- **Integração com Redes Sociais:** É bastante comum que as empresas possuam um perfil nas redes sociais. Geralmente esta é uma medida de *marketing* e aproximação o público. Estes perfis tendem a ser utilizados para divulgação de novos produtos e novidades da empresa.

Além de seus perfis oficiais, as empresas podem adotar um perfil secundário ou página como canal de suporte. É bastante comum que as plataformas de redes sociais

forneçam ferramentas específicas para empresas, sejam elas gratuitas ou não. O ponto interessante no uso das redes sociais como canais de suporte está no fato de que sequer há uma necessidade de recrutamento da comunidade. Os próprios usuários da rede social são capazes de fornecer uma resposta muito rapidamente.

Para se realizar a integração direta com redes sociais, é necessário utilizar o conjunto de APIs fornecidas pela própria rede social. Isso permite que o sistema seja capaz de identificar requisições de suporte provenientes deste meio. Além disso, essa integração também permite computar as colaborações que um parceiro poderia realizar diretamente na rede social, sem necessariamente, utilizar a plataforma desenvolvida.

3.3.2 Requisitos não Funcionais

Os requisitos não funcionais não estão ligados diretamente às funções utilizadas pelo usuário. Geralmente estes requisitos especificam restrições ou propriedades do *software*, como por exemplo, limite de espaço ocupado em disco e tempo de resposta, respectivamente (SOMMERVILLE, 2007). Os seguintes requisitos não funcionais foram levantados para o projeto:

- **Facilidade de uso:** Fidelizar um colaborador na plataforma é algo muito importante para mantê-la relevante. Mas para isso, é importante que a plataforma em si seja suficientemente simples para que qualquer usuário destreinado consiga aprender rapidamente como utilizá-la.
- **Desempenho:** *Softwares* com baixo tempo de resposta, ou com travamentos, são causas conhecidas de frustrações de usuários. A plataforma aqui proposta tenta oferecer uma navegação mais fluída. Para isso, as computações são executadas em *Threads* separadas das *Threads* responsáveis pela interface gráfica.

Fazendo-se este tratamento nas aplicações clientes, o tempo de resposta fica limitado ao tempo de processamento no lado do servidor, ao tempo de tráfego da resposta na rede de Internet, e ao tempo de interpretação da resposta na aplicação cliente.

- **Segurança:** É muito importante que as informações tanto dos colaboradores e funcionários quanto dos usuários do suporte estejam protegidas. A primeira medida básica é utilizar o protocolo SSL sobre as requisições de dados HTTP entre clientes e servidor. Esta medida utiliza transferência de pacotes de redes criptografados, tornando o sistema menos susceptível ao vazamento de dados.

As informações que possam violar a segurança dos usuários também devem ser armazenadas de forma criptografada pelo servidor. Além disso, é importante que

o servidor mantenha a chave decriptografia armazenada em um alto nível de segurança.

3.4 Padrões de Projeto

Durante o desenvolvimento de *software* é importante utilizar de boas práticas que tornem o código mais simples e organizado, além de garantir um maior nível de manutenibilidade. Isso significa que o *software* desenvolvido deve ser suficientemente compreensível para que outros possam editá-lo. Logo, o *software* também deve ser estruturado de tal forma que sua manutenção seja simples, ou seja, a modificação de um recurso não deve impactar nas demais funcionalidades.

Os padrões de projeto são metodologias de *software* desenvolvidas pensando em resolver determinado problema de *software* (PRESSMAN; MAXIM, 2016). Existem padrões de uso mais específico a padrões capazes de estruturar todo um projeto. É importante salientar que os padrões de projeto não são capazes de solucionar todos os tipos de problemas, e sua escolha deve ser feita baseada nos requisitos do projeto. Nas sub-seções a seguir serão abordados os principais padrões de projeto empregados no desenvolvimento deste trabalho.

3.4.1 MVVM

Criado em 2005 por John Gossman, um dos arquitetos da Microsoft responsável pelo *.net Framework*, WPF e *Silverlight* (SMITH, 2009). O MVVM é um padrão de projeto de organização da interface de usuário. Em outras palavras, ele é capaz de definir a forma com que a lógica dos componentes gráficos da aplicação serão estruturados no projeto. Ele consiste em dividir o desenvolvimento em 3 camadas distintas.

A primeira camada, chamada de *View*, é responsável por manter o design de todos os componentes e layout das interfaces gráficas. Para projetos como WPF, *Silverlight* e UWP, esta camada é geralmente desenvolvida utilizando-se XAML. O XAML é uma extensão do XML criada pela Microsoft para o desenvolvimento de interfaces gráficas. As interfaces gráficas construídas com XAML para o padrão MVVM não possuem lógica de negócios, fazendo referências de conexões com os dados da camada *ViewModel* (SMITH, 2009).

A segunda camada, chamada de *Model*, é responsável por modelar e abstrair a estrutura dos dados utilizada na interface de usuário. Nesta camada os dados são modelados em classes simplificadas sem nenhuma lógica de negócios, ou seja, sem nenhum método ou código que faça uso direto dos dados modelados. Esta tarefa fica a cargo da terceira camada, a *ViewModel*.

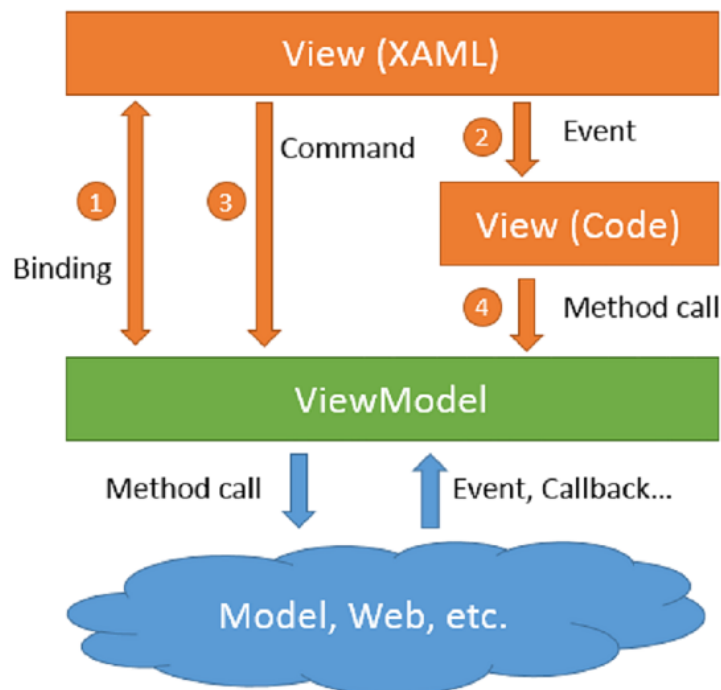


Figura 9 – Diagrama da estrutura do padrão de projeto MVVM.

Fonte: Microsoft MSDN Magazine Issues.

A terceira camada, denominada *ViewModel*, é onde toda a lógica de negócios empregada na interface é realizada. A *ViewModel* é um objeto instanciado na camada de apresentação capaz de reagir às interações do usuário com a interface gráfica. Para isso, ligações dinâmicas de dados, chamadas na plataforma de *Data Bindings*, oferecem uma forma de detecção de alterações nos estados dos componentes. Assim, a *ViewModel* é capaz de fornecer à camada *View*, dados atualizados de acordo com a estrutura definida na camada *Model*.

De um modo geral, o padrão de projeto MVVM pode ser facilmente comparado ao padrão MVC, comumente utilizado no desenvolvimento de aplicativos para a *web*. Assim como o MVVM, o MVC faz uma separação da lógica de apresentação em três camadas distintas: *Model*, *View* e *Controller* (LEFF; RAYFIELD, 2001). A estrutura e fluxo de dados do padrão de projeto MVVM pode ser observado na Figura 9 (BUGNION, 2013).

Embora o MVVM não seja um padrão de projeto capaz de ser aplicado em todos os casos, este trabalho empregou este padrão em todas as páginas de apresentação para o usuário. Mesmo o MVVM sendo utilizado na camada de apresentação, isso não exclui a possibilidade de uso de outros padrões de projeto juntamente a ele. As tarefas executadas fora da camada de apresentação, por exemplo, utilizaram diferentes padrões de projeto em sua construção.

3.4.2 State

Em combinação com o MVVM, o padrão de projeto *State* permite notificar para a camada de apresentação, mudanças nos estados dos objetos de dados da camada *Model*. Isso permite, por exemplo, atualizar a interface gráfica do usuário com atualizações de dados provenientes do servidor. Para isso, no projeto desenvolvido, um serviço em segundo plano é utilizado para procurar notificações no servidor de dados.

3.4.3 Command

Este também é um padrão de projeto utilizado em conjunto com o MVVM. Ele permite que a camada *ViewModel* seja notificada acerca das interações do usuário com os controles da interface gráfica, através de comandos. Desta forma a camada *ViewModel* é capaz de realizar tarefas correspondentes às requisições feitas pelo usuário, fornecendo atualizações para a interface gráfica. A Figura 10 ilustra um exemplo de aplicação do padrão de projeto *Command* com ligação dinâmica entre *View* e *ViewModel* utilizado no *software* desenvolvido.

```
<ct:LoadBox
  IsLoading="{x:Bind ViewModel.IsThreadsLoading, Mode=OneWay}"
  HasNetworkError="False"
  HasProcessingError="{x:Bind ViewModel.ThreadsLoadFailed,Mode=OneWay}"
  ReloadCommand="{x:Bind ViewModel.ReloadThreadsCommand}"
  DataContext="{x:Bind ViewModel}"
  MinHeight="250">
```

Figura 10 – Exemplo de utilização do padrão de projeto *Command* juntamente ao padrão MVVM.

3.4.4 MVC

O padrão de projeto MVC é comumente utilizado na construção de aplicações para a WEB. Este padrão estrutura a construção do *software* em três partes distintas, *View*, *Model* e *Controller* (LEFF; RAYFIELD, 2001).

A *View* e a *Model*, possuem mesma funcionalidade das partes de mesmo nome do padrão MVVM anteriormente citado. No caso da *View*, a estrutura da interface gráfica é geralmente escrita utilizando HTML, ao invés de XAML, como no MVVM. Para aplicativos da WEB baseados no padrão MVC para a construção de APIs, a *View* geralmente não é utilizada, sendo substituída pela estruturação JSON ou XML dos objetos de dados baseados na *Model*.

A *Model* geralmente é escrita para refletir a estrutura do banco de dados. Ferramentas e *frameworks*, como mapeamento objeto-relacional, podem ser utilizados para fazer uma ponte entre o banco de dados e a estrutura especificada na *Model*. Isso permite

recuperar dados estruturados de acordo com os padrões da linguagem de programação utilizada.

O *Controller* é responsável por disponibilizar os pontos de acesso para a WEB. No *Controller* são recebidas as requisições do usuário, sendo processadas para entregar uma resposta baseada na estrutura gráfica definida na *View*, ou na estrutura JSON ou XML dos dados definidos na *Model*. A Figura 11 apresenta a estrutura e modelo de requisição HTTP do padrão MVC (ANDERSON; WASSON, 2019).

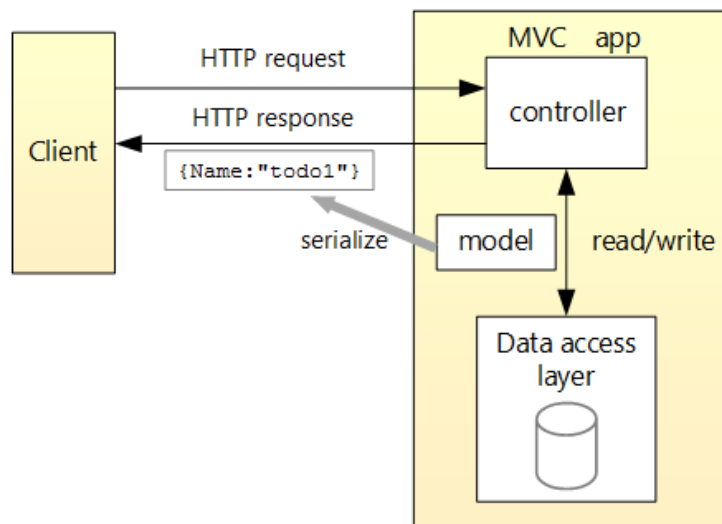


Figura 11 – Diagrama da estrutura do padrão de projeto MVC.

Fonte: Microsoft Corporation.

3.5 Reuso de *software*

Para Sommerville (SOMMERVILLE, 2007), a engenharia de *software* baseada em reuso é uma estratégia que procura maximizar o reuso de código, sistemas e componentes de uma aplicação.

Durante o desenvolvimento do projeto aqui proposto, houve uma atenção voltada para a generalização das principais áreas do *software*. Em projetos da plataforma universal do *Windows*, existe a possibilidade de criação de bibliotecas reutilizáveis em uma ampla gama de produtos da Microsoft. As bibliotecas de vínculo dinâmico, popularmente conhecidas pela extensão de arquivo *.dll*, são capazes de abrigar todo um conjunto de classes que compõe uma biblioteca. Além disso, as "*dlls*" são capazes de fazer referência para outras "*dlls*", permitindo uma melhor separação de bibliotecas de acordo com suas funções.

Toda a camada de comunicação entre a aplicação cliente e o servidor foi separada em uma biblioteca específica para a comunicação e transferência de dados entre as partes. Essa mesma biblioteca criada para tratar essa comunicação no projeto da plataforma universal do *Windows* poderá ser reutilizada no desenvolvimento de aplicativos para as plataformas

móveis *Android*² e *IOS*³, pertencentes à Google LLC. e Apple Inc., respectivamente. Isso é possível graças à plataforma *Xamarin*⁴, da Microsoft, para desenvolvimento de aplicações móveis, que permite o reuso simplificado destas bibliotecas.

3.6 Desenvolvimento do Servidor

Para desenvolver a aplicação do servidor, o padrão de projeto MVC foi utilizado para definir a estrutura e organização dos arquivos e classes do projeto. A tecnologia escolhida para executar a aplicação foi o *Asp.net Core* da Microsoft Corporation (SMITH, 2019).

Uma similaridade direta entre o padrão MVC e o MVVM, é a *Model*, que está presente em ambos os padrões. Isso permite utilizar os mesmos conjuntos de classes que modelam os dados tanto no lado da aplicação do servidor quanto no lado do aplicativo cliente.

Para o banco de dados foi utilizado o *framework* de mapeamento objeto-relacional, *Entity Framework*, também da Microsoft Corporation. Através do recurso denominado *Code-First*, é possível gerar a estrutura do banco de dados a partir da modelagem dos dados da *Model*. O *Visual Studio* permite a criação automática dos controladores do padrão MVC de forma automática, através de um formulário onde são informados os modelos dos dados e a classe do contexto do banco de dados. Esta geração automática inclui métodos para acessar, editar, excluir ou efetuar consultas no banco de dados. Além disso, os métodos gerados automaticamente também respeitam o tipo adequado de requisição HTTP baseado no tipo de ação, como *GET*, *POST*, *PUT*, *DELETE*, etc. A resposta do servidor é entregue ao cliente no formato JSON (ANDERSON; WASSON, 2019).

3.7 Testes de *Software*

Para Pressman, testar um *software* tem por objetivo revelar os erros cometidos inadvertidamente durante as etapas de projeto e construção do produto (PRESSMAN; MAXIM, 2016).

O teste de *software* é uma importante etapa do processo de desenvolvimento e deve ser visto como uma forma de agregar qualidade ao produto final. *Softwares* defeituosos são capazes de prover grandes frustrações aos seus usuários. Em certos casos, o usuário poderá buscar por produtos similares, para utilizar a mesma funcionalidade que o *software* propõe.

² <http://www.android.com>

³ <https://www.apple.com/ios/>

⁴ <http://www.xamarin.com>

Os testes de *software* devem ser cuidadosamente planejados. É importante que o teste seja suficientemente abrangente para maximizar a descoberta de erros. Porém, despender esforços no planejamento e execução de testes demasiadamente complexos pode trazer custos desnecessários, podendo ainda assim, deixar passar erros durante o processo (PRESSMAN; MAXIM, 2016).

3.7.1 Testes de Unidade

Os testes de unidade direcionam os esforços para testar a menor unidade do projeto de *software*, o componente ou módulo de *software* (PRESSMAN; MAXIM, 2016). Em linguagens orientadas a objetos, como o C#, a menor unidade tratada pelo teste é a Classe.

Em um projeto da plataforma universal do *Windows*, é possível realizar a criação de classes com rotinas específicas de teste. Essas classes fazem a instanciação das classes alvo do teste. Cada um dos construtores, métodos, propriedades e eventos devem ser testados para um universo de possibilidades definidos no projeto do teste. Este universo de possibilidades especifica um conjunto de caminhos discretos com o qual o *software* normalmente percorreria em uma execução normal. É importante lembrar que um conjunto de possibilidades muito grande, pode tornar a etapa de teste demorada e dispendiosa para a equipe de desenvolvimento. Por este motivo, os testes de unidade devem ser cuidadosamente pensados para serem suficientemente abrangentes e com custo relativamente aceitável.

Na Figura 12, temos um método de teste destinado a verificar o comportamento da classe responsável pelas notificações do aplicativo proposto. Note pela imagem que até o lançamento de uma exceção de código para uma determinada circunstância deve ser testado, principalmente se este for o comportamento programado para essa situação. Neste exemplo, a invocação do método *GetAsync* com parâmetro igual a nulo deverá lançar uma exceção do tipo *NullReferenceException*.

```
[TestClass]
public class NotificationTests
{
    [TestMethod]
    public void NotificationInvalidRequestTest()
    {
        Assert.ThrowsException<NullReferenceException>(async () =>
        {
            //test invalid method parameter
            await Notifications.GetAsync(null).ConfigureAwait(false);
        });
    }
}
```

Figura 12 – Exemplo de código de teste escrito para verificar o comportamento do método *GetAsync* para um parâmetro inválido.

4 Resultados

Como a camada do lado do servidor é composta por APIs destinadas ao consumo das aplicações clientes, os resultados apresentados neste capítulo terão foco nestas aplicações. A principal motivação se deve à maior facilidade para demonstrar as funcionalidades graficamente. É importante lembrar que todas as operações de dados, como inserção, alteração, leitura ou exclusão são inteiramente executadas na camada do servidor, através de comandos de API requisitados pelas aplicações clientes.

4.1 Design

O design do aplicativo desenvolvido empregou uma paleta de cores escura. Essa escolha contribui para economia de energia no dispositivo para telas OLED, já que reduz o nível de brilho necessário para exibir os gráficos (LUDWICKI; BOROSON; SPINDLER, 2009). Além disso, o aplicativo procura utilizar cores sólidas sempre que possível. Esta medida também contribui para economia de energia uma vez que menos cálculos são necessários ao processador gráfico para desenhar as telas.

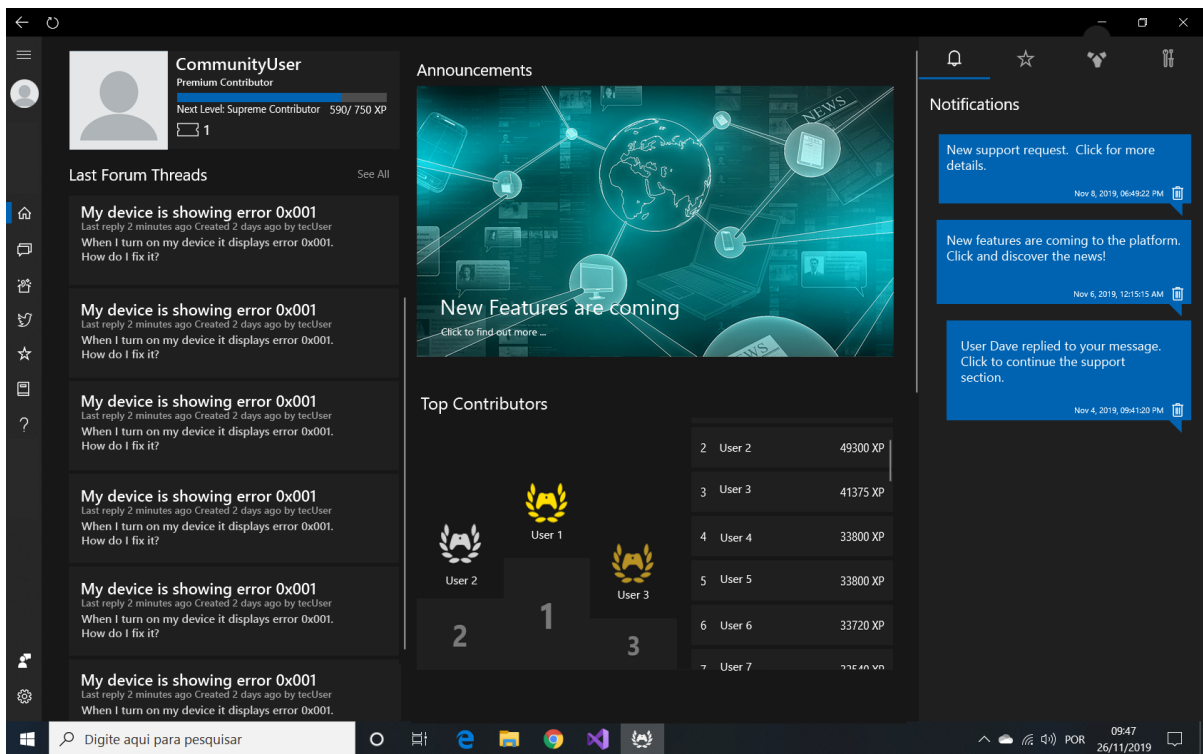


Figura 13 – Tela inicial do aplicativo.

Na Figura 13 podemos observar o layout da tela inicial do aplicativo. Na lateral esquerda há uma barra de menu com acesso a todas as áreas do aplicativo. Na tela

inicial também há um conjunto de informações úteis para o usuário, como as postagens mais recentes no fórum, anúncios, resumo do perfil, notificações, e algumas ferramentas adicionais posicionadas na lateral esquerda. Também há um *podium* posicionando os usuários colaboradores da plataforma em uma escala crescente baseado no número de pontos de experiência conquistados por suas ações dentro da plataforma.

Na barra superior do aplicativo, no lado esquerdo, existem dois botões. Estes dois botões estão presentes em todas as telas do aplicativo. O primeiro, identificado por uma seta direcionada para a esquerda, é responsável pela navegação reversa entre as páginas do aplicativo, ou seja, voltar para as páginas anteriormente acessadas.

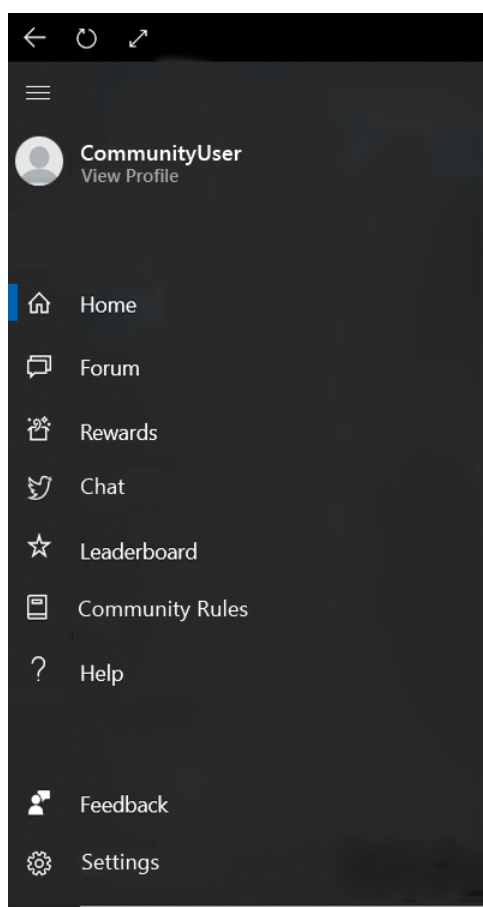


Figura 14 – Menu de opções lateral do aplicativo.

O segundo botão, identificado por uma seta em um semi-círculo, é responsável por atualizar os dados da página que está sendo exibida. Por exemplo, na tela inicial (Figura 13), o trecho "*Last Forum Threads*", responsável pelos tópicos mais recentes do fórum; o trecho "*Top Contributors*", responsável pelo *ranking* de colaboradores; o trecho "*Announcements*", responsável pelos anúncios da plataforma; e o trecho "*Notifications*" responsável pelas notificações ao usuários, serão todos atualizados fazendo-se novas requisições de dados para à API no servidor quando clicado.

Um recurso também presente na tela inicial do aplicativo é a atualização automática

individual de cada trecho exibido. Para os trechos mais importantes, como o fórum e notificações, um temporizador foi configurado de modo a efetuar requisições de dados atualizados de forma automática em intervalos pré definidos de tempo. Demais trechos como anúncios e *ranking* não foram configurados para se auto atualizarem. Esta medida tem por objetivo reduzir o número de requisições feitas para o servidor, de forma a priorizar os dados mais importantes para o colaborador.

O menu lateral expansível, demonstrado na Figura 14, contém um acesso rápido para cada uma das sub-páginas do aplicativo. A barra lateral possui um indicador em azul responsável por indicar qual é a atual página aberta no aplicativo. Esse indicador muda dinamicamente à medida que o colaborador navega pelos menus, ou quando o colaborador retrocede na navegação utilizando o botão voltar na barra de título.

4.2 Perfil do colaborador

A página do perfil do usuário colaborador, Figura 15, possui uma considerável importância já que ela é responsável por apresentar para o colaborador o seu desempenho. Nesta página o colaborador pode acompanhar o seu progresso através dos níveis definidos no programa, além dos valores necessários de pontos de experiência para avançar dentro da plataforma.

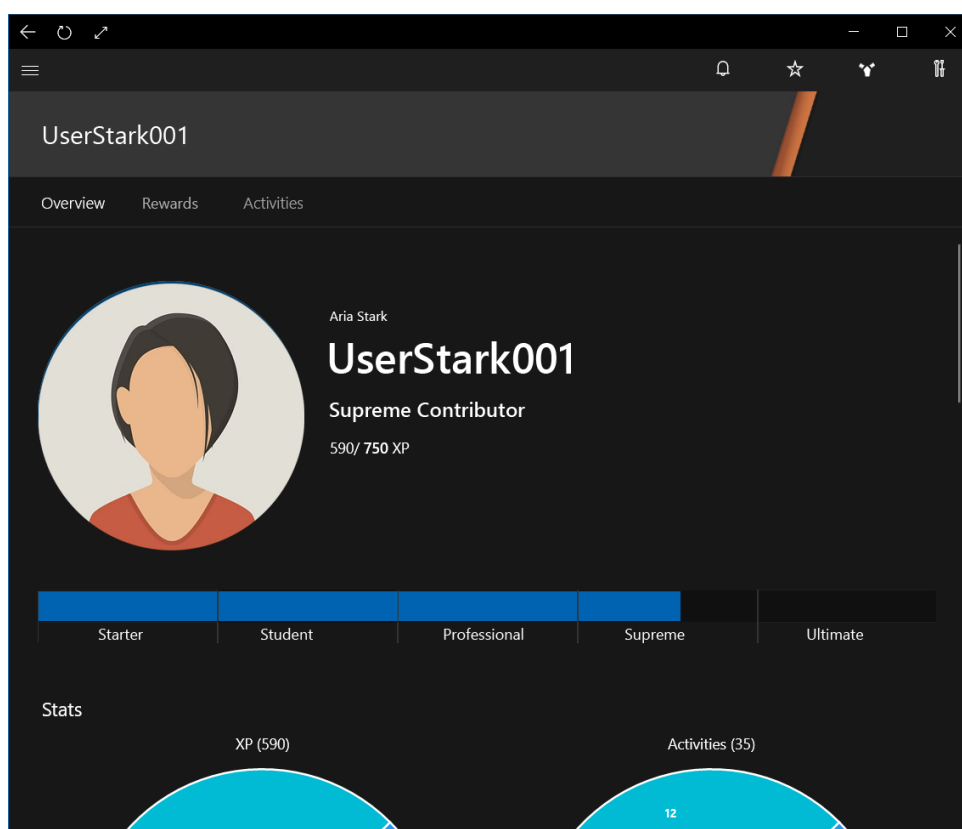


Figura 15 – Tela do perfil do usuário colaborador.

Também é possível para o colaborador acompanhar e visualizar as recompensas adquiridas através do menu "*Rewards*". Neste menu o colaborador pode resgatar os artefatos conquistados aplicando-os ao seu perfil, no caso de itens digitais, ou informando o endereço para postagem, no caso de recompensas físicas.

O menu "*Activities*" é responsável por exibir ao colaborador todas as ações por ele realizadas além dos valores de pontos de experiência atribuídos. Este menu é uma forma do colaborador acompanhar a contabilização de suas contribuições além de permitir descobrir quais ações são capazes de prover maiores recompensas.

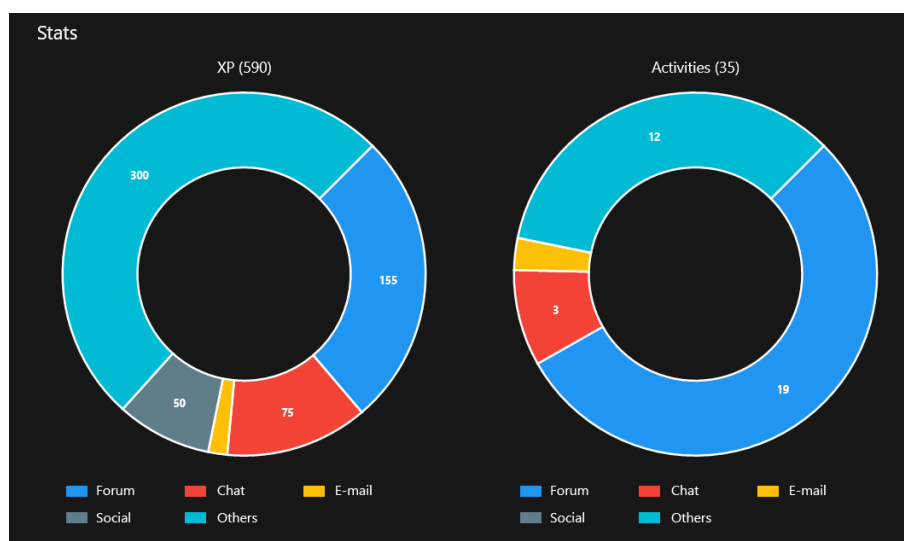


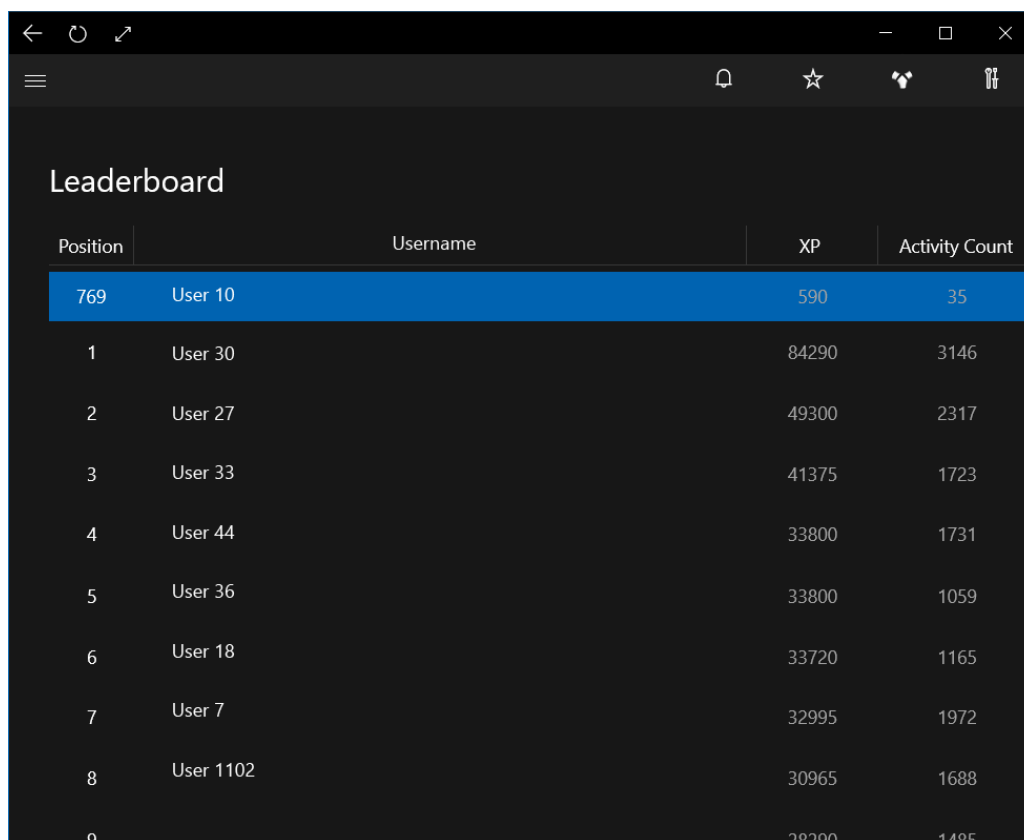
Figura 16 – Tela de Estatísticas do usuário colaborador.

A página do perfil do usuário também apresenta alguns gráficos de desempenho para o colaborador, como na Figura 16. Estes gráficos permitem que o colaborador acompanhe sua evolução à medida que suas ações são computadas dentro da plataforma. Também é possível visualizar não somente o quantitativo de atividades, mas também o quantitativo de pontos de experiência arrecadados por cada uma das atividades individualmente.

4.3 Placares de Líderes

O *ranking* de usuários colaboradores, apresentado na Figura 17 ("*Leaderboard*"), exibe um quadro com todos os colaboradores participantes da comunidade e seu respectivo desempenho perante aos demais colaboradores. Além de permitir que o colaborador visualize sua posição na plataforma, esse quadro também possui um efeito implícito: competição.

O *ranking* de usuários colaboradores pode ser capaz de promover uma competição saudável entre os próprios colaboradores da plataforma. A tendência é que alguns colaboradores tentem alcançar as primeiras posições do *ranking* enquanto os primeiros colocados



Position	Username	XP	Activity Count
769	User 10	590	35
1	User 30	84290	3146
2	User 27	49300	2317
3	User 33	41375	1723
4	User 44	33800	1731
5	User 36	33800	1059
6	User 18	33720	1165
7	User 7	32995	1972
8	User 1102	30965	1688
9		28290	1485

Figura 17 – Placares de líderes da comunidade.

tentarão mantê-las. Este é um cenário positivo, onde haverá uma competição pelos chamados de suporte dentro da plataforma. Isso significa que para o cliente apresentando problemas, a chance de receber uma resposta em menor tempo será maior.

4.4 Mensagens Instantâneas

O *chat*, Figura 18, é talvez um dos principais canais de suporte da plataforma. Ele permite que um usuário com problemas se comunique em tempo real com um colaborador ou funcionário do suporte oficial. O menu *"Available Threads"* exibe para o colaborador todas as requisições de suporte que ainda não foram aceitas por algum colaborador. O colaborador tem acesso a uma breve e resumida descrição da solicitação para apoiar sua decisão de aceitar a requisição ou não. Quando uma requisição é aceita por um colaborador, através de um sistema de notificação em tempo real, a requisição é removida para todos os demais colaboradores.

O menu *"Active Threads"* permite que o colaborador visualize todas as requisições de suporte que ele está atualmente atendendo. É importante salientar que um colaborador pode aceitar mais de uma requisição de suporte por vez. Caso um colaborador aceite uma requisição, mas decida abandoná-la, esta requisição sairá do menu *"Active Threads"* do colaborador e reaparecerá para todos os demais colaboradores no menu *"Available*

Threads". Isso permite que outro colaborador aceite a requisição.

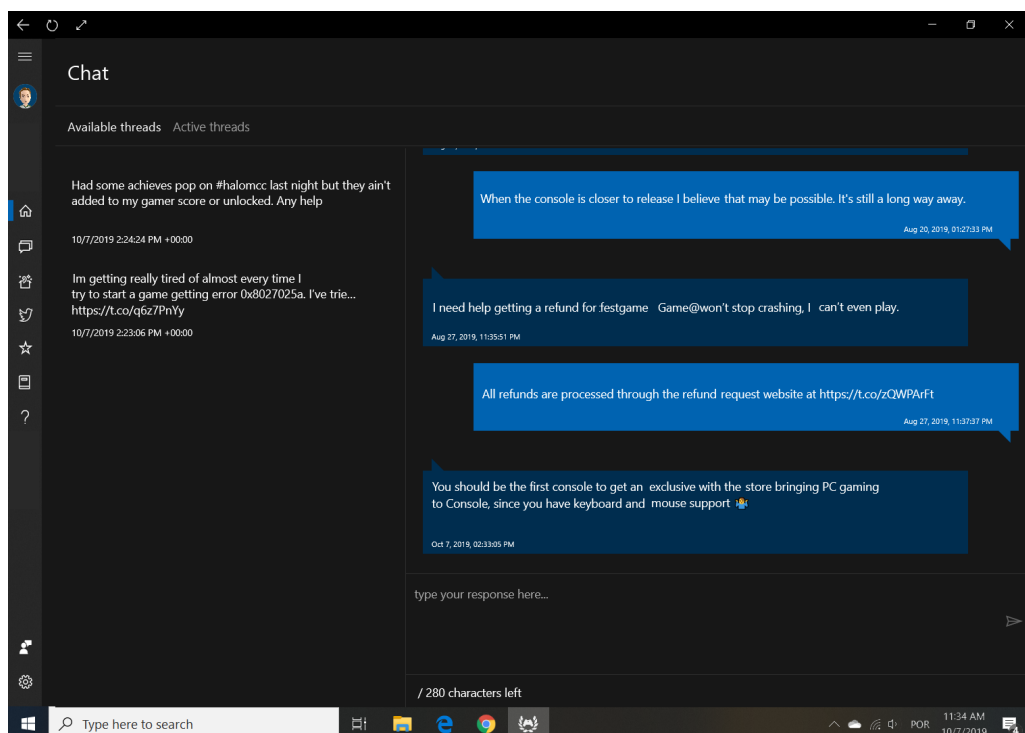


Figura 18 – Tela de mensagens instantâneas com o usuário.

Quando a requisição de suporte é aceita, uma tela com o histórico de mensagens é aberta na parte lateral direita da página. É possível visualizar o histórico completo de mensagens, incluindo mensagens de outros colaboradores que participaram do suporte anteriormente. Visualizar o histórico permite ao atendente compreender o contexto da solicitação mais facilmente.

4.5 Fórum de Discussão

O fórum de discussão é a parte fundamental da plataforma capaz de permitir que qualquer usuário, seja colaborador ou não, possa ajudar a outros usuários que estejam com algum problema. Para isso, a plataforma permite que qualquer pessoa possa visualizar e responder a postagens a respeito de problemas por parte de determinado usuário.

O *design* desta área é dividido em duas colunas. No lado esquerdo, há uma listagem com todos os tópicos de discussão em aberto por ordem cronológica decrescente. Esta listagem também é equipada com um algoritmo de carregamento dinâmico. Isso faz com que o aplicativo faça o carregamento de um número limitado de postagens inicialmente. Conforme o usuário faz a rolagem da lista, novas postagens são requisitadas para o servidor e adicionadas à lista.

No lado direito da tela são abertas as postagens que o colaborador seleciona na listagem. Nesta parte do fórum, o colaborador pode fazer a rolagem da página e visualizar

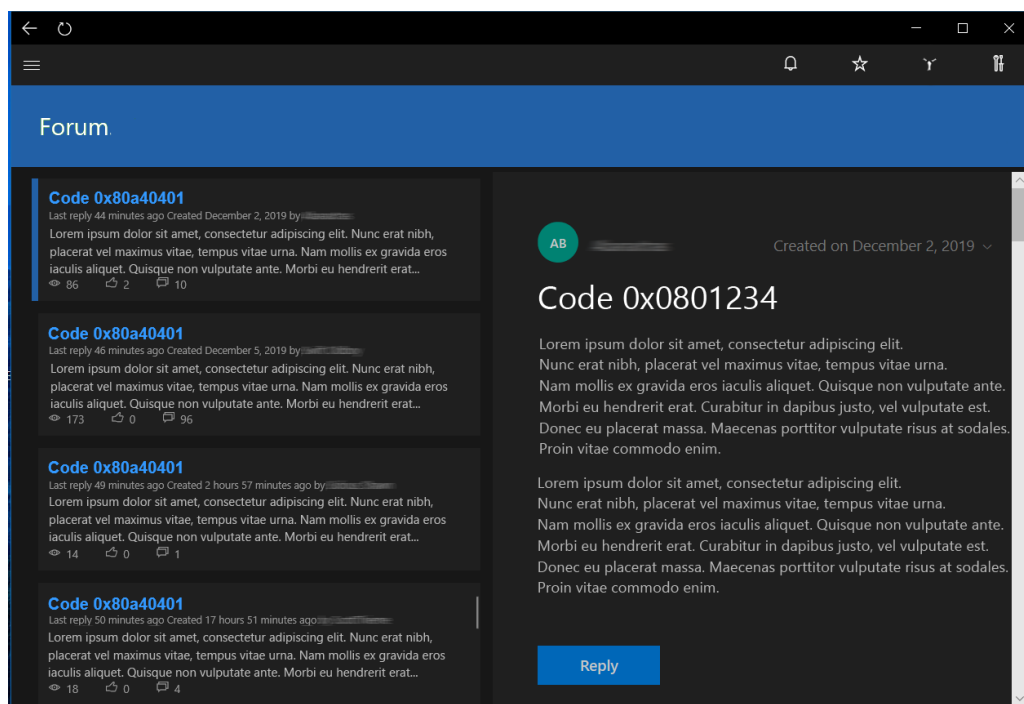


Figura 19 – Tela correspondente ao recurso de Fórum integrado ao aplicativo.

respostas de outros usuários além de também incluir sua própria resposta.

4.6 Sistema de Notificação

É importante que o colaborador seja notificado quando houver novas atividades disponíveis para ele. Na Figura 20, podemos ver um exemplo de notificação emitida pelo aplicativo quando uma nova requisição de suporte é feita por um usuário da comunidade.

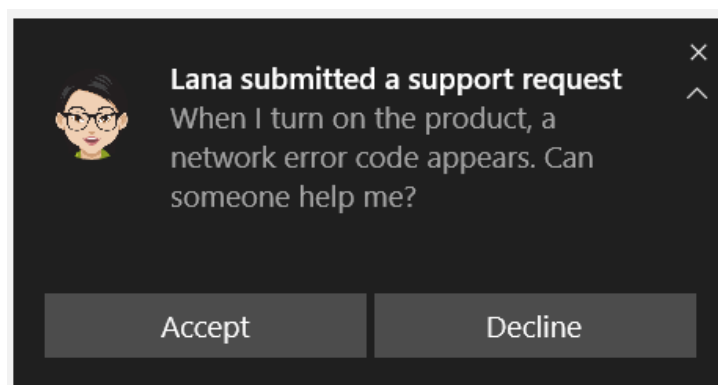


Figura 20 – Exemplo de notificação emitida pelo aplicativo para um novo evento.

Este recurso evita que o colaborador tenha que ficar atualizando constantemente as páginas do aplicativo em busca de novas atividades. Além disso, o sistema de notificações é capaz de emitir alertas mesmo quando o aplicativo está fechado. Isso é possível devido a

um processo registrado para execução em segundo plano, responsável por procurar por notificações.

4.7 Ferramentas de apoio ao suporte

Pressman define **requisito fascinante**, os requisitos que estão além das expectativas do usuário (PRESSMAN; MAXIM, 2016). Embora não seja necessária para se prestar um atendimento ao usuário, a ferramenta de busca de código de erros e informações, descrita na Figura 21, é uma funcionalidade não esperada pelo usuário. Ela consiste de um banco de dados de informações de suporte mantido na nuvem, com acesso disponibilizado via API.

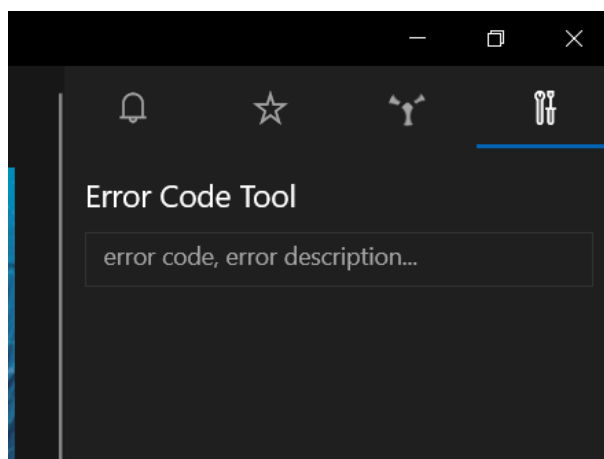


Figura 21 – Ferramenta de busca de códigos de erro e páginas de suporte.

Esta ferramenta de princípio simples, utiliza um algoritmo de busca de texto completo para tentar encontrar a melhor resposta para determinada pesquisa (SUBRAMANIAM et al., 2004). Este algoritmo, presente em alguns sistemas de banco de dados, pode ser comparado ao que buscadores, como o Google¹, por exemplo, fazem para disponibilizar resultados de busca.

4.8 Tratamento de erros na interface gráfica

Quando por algum motivo o aplicativo não é capaz de realizar alguma tarefa, uma mensagem é exibida para o usuário detalhando o problema e como possivelmente resolvê-lo. Na Figura 22 temos um exemplo de erro de conexão exibido para o usuário. Note que embora constituído de duas frases simples, elas são suficientemente claras em descrever o problema e uma possível solução.

Um botão capaz de permitir que o usuário tente novamente realizar a ação também é exibido nos diálogos. No caso de erros de conexão de rede, o aplicativo é capaz de detectar

¹ <http://www.google.com>

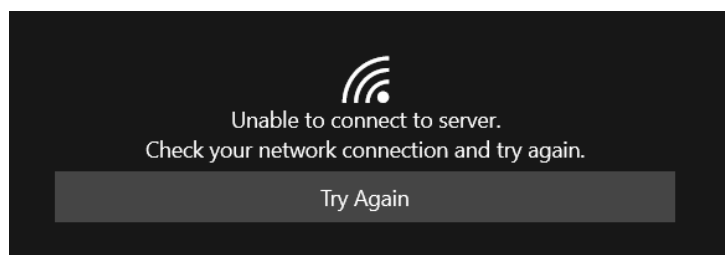


Figura 22 – Exemplo de diálogo exibido para o usuário na ocorrência de um erro de rede.

automaticamente mudanças no adaptador de rede e tentar efetuar a tarefa novamente de forma automática. Nos casos em que o problema não for local, como por exemplo, servidor *off-line*, o usuário poderá utilizar o botão *Try Again* até que uma nova conexão possa ser reestabelecida.

4.9 Coletando o *Feedback* do Usuário

Para coletar e ouvir o feedback dos usuários, o aplicativo utiliza serviços integrados à loja do *Windows*. Quando a opção *feedback*, presente no menu lateral do aplicativo é acionada, o aplicativo *Hub de Feedback*, nativo nas versões mais recentes do *Windows* é aberto. Neste aplicativo o usuário pode enviar seus comentários e opiniões.

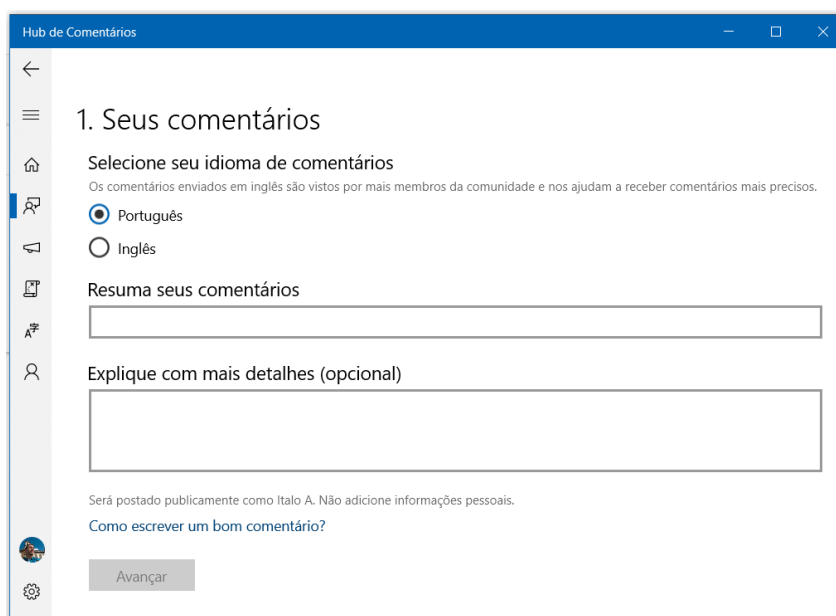


Figura 23 – Captura de tela do *Hub* de comentários dos serviços da loja do *Windows* e *XBOX*.

Os comentários e opiniões enviados pelo *Hub de feedback* podem ser lidos posteriormente através do painel do desenvolvedor da loja do *Windows*. Note que o *Hub de feedback*, diferentemente do aplicativo desenvolvido, é multilíngue, ou seja, será apresentado para o usuário no idioma configurado em seu sistema operacional.

5 Conclusão

O atendimento de qualidade aos clientes deve ser visto pelas empresas como uma forma de agregar qualidade à marca. Um bom atendimento ao cliente é capaz de promover a sensação de segurança do usuário com relação à um produto ou serviço. A certeza do cliente com relação à solução de suas dúvidas e problemas é possivelmente capaz de fidelizar o usuário juntamente à marca.

Neste sentido, há uma necessidade de reconhecimento por parte do meio corporativo, às possíveis contribuições da comunidade usuária de seus próprios produtos. É preciso perceber os usuários que se destacam por promover ações positivas e sanando os problemas de muitos usuários.

A criação de plataformas que encorajam os usuários a se expressarem livremente pode ser vista como uma forma de contornar a crescente demanda por suporte à medida em que o produto ou serviço se populariza.

Neste trabalho, a plataforma de suporte desenvolvida procurou adotar as principais ferramentas baseadas em comunidade, tomando como base os principais casos de sucesso relatados no mercado. Além disso, a plataforma proposta procurou suprir necessidades e contornar problemas relatados por usuários de plataformas já existentes.

Por fim, o trabalho proposto também procurou criar uma experiência multi dispositivos, onde o conjunto de APIs disponibilizadas pelo servidor, e as bibliotecas reutilizáveis desenvolvidas, permitem desenvolver novos clientes para novos dispositivos em menor tempo. Tudo isso para permitir que os colaboradores possam enviar suas contribuições de onde estiverem.

5.1 Trabalhos Futuros

O próximo passo na continuação deste trabalho é justamente realizar o desenvolvimento de novos clientes para outros dispositivos e sistemas operacionais. A plataforma Xamarin permite o reaproveitamento da biblioteca de comunicação desenvolvida na criação de aplicativos para Android de IOS.

Há também a possibilidade de um estudo em cima de novos dispositivos, não inicialmente pensados, que possam ser capazes de funcionar como plataforma de apoio. Este é o caso das *Smart TVs*, capazes de executar aplicativos de sua loja própria.

O *feedback* coletado do usuário também permitirá evoluir as funcionalidades implementadas além de possivelmente serem desenvolvidas novas.

Referências

- ACTIVISION. *Frequently Asked Questions about the Activision Ambassador Program*. 2019. Disponível em: <https://support.activision.com/articles/en_US/FAQ/Activision-Support-Ambassador-FAQ>. Citado na página 21.
- AMINE, A.; SITZ, L. How does a virtual brand community emerge? some implications for marketing research. 01 2004. Citado na página 20.
- ANDERSON, R.; WASSON, M. *Tutorial: Create a web API with ASP.NET Core*. 2019. Acesso em: 09 dez. 2019. Disponível em: <<https://docs.microsoft.com/en-us/aspnet/core/tutorials/first-web-api?view=aspnetcore-3.1&tabs=visual-studio>>. Citado 2 vezes nas páginas 38 e 39.
- BNL. *The First Video Game?* 2013. Disponível em: <<https://www.bnl.gov/about/history/firstvideo.php>>. Citado na página 16.
- BUGNION, L. Mvvm - messenger and view services in mvvm. *MSDN Magazine Issues*, v. 28, n. 3, 2013. Acesso em: 02 dez. 2019. Disponível em: <<https://docs.microsoft.com/en-us/archive/msdn-magazine/2013/march/mvvm-messenger-and-view-services-in-mvvm>>. Citado na página 36.
- DELAMARO, M.; JINO, M.; MALDONADO, J. *Introdução ao teste de software*. [S.l.]: Elsevier Brasil, 2013. Citado 3 vezes nas páginas 16, 25 e 27.
- ECMA INTERNATIONAL. *C# Language Specification*. 2017. Acesso em: 02 dez. 2019. Disponível em: <<https://www.ecma-international.org/publications/files/ECMA-ST/ECMA-334.pdf>>. Citado na página 31.
- ELECTRONIC ARTS. *Sobre o Answers HQ*. 2019. Acesso em: 05 dez. 2019. Disponível em: <<https://answers.ea.com/t5/help/faqpage>>. Citado 2 vezes nas páginas 24 e 25.
- LEFF, A.; RAYFIELD, J. T. Web-application development using the model/view/controller design pattern. In: IEEE. *Proceedings fifth ieee international enterprise distributed object computing conference*. [S.l.], 2001. Citado 2 vezes nas páginas 36 e 37.
- LUDWICKI, J. E.; BOROSON, M. L.; SPINDLER, J. P. *OLED device with improved power consumption*. [S.l.]: Google Patents, 2009. US Patent 7,510,454. Citado na página 41.
- MCALEXANDER, J.; SCHOUTEN, J.; KOENIG, H. Building brand community. *Journal of Marketing*, v. 66, 01 2002. Citado na página 20.
- MICROSOFT CORPORATION. *Xbox Ambassadors*. 2011. Acesso em: 9 set. 2019. Disponível em: <<https://ambassadors.microsoft.com/xbox>>. Citado na página 22.
- MICROSOFT CORPORATION. *Support contact info*. 2018. Acesso em: 02 dez. 2019. Disponível em: <<https://docs.microsoft.com/en-us/windows/uwp/publish/enter-app-properties#support-contact-info>>. Citado na página 20.

- MICROSOFT CORPORATION. *Ambassador App*. 2019. Disponível em: <<https://answers.microsoft.com/en-us/xbambassadors/forum/>>. Citado na página 24.
- MICROSOFT CORPORATION. *The Ambassador Handbook*. 2019. Acesso em: 9 set. 2019. Disponível em: <<https://ambassadors.microsoft.com/xbox/handbook>>. Citado 2 vezes nas páginas 22 e 23.
- MICROSOFT CORPORATION. *Criar APIs Web com o ASP.NET Core*. 2019. Disponível em: <<https://docs.microsoft.com/pt-br/aspnet/core/web-api/?view=aspnetcore-3.0>>. Citado na página 30.
- MICROSOFT CORPORATION. *Desenvolvimento na Plataforma Universal do Windows*. 2019. Disponível em: <<https://visualstudio.microsoft.com/pt-br/vs/features/universal-windows-platform/>>. Citado na página 32.
- MICROSOFT CORPORATION. *Get the Deets: The Xbox Ambassadors Experience*. 2019. Acesso em: 02 dez. 2019. Disponível em: <<https://blog-ambassadors.xbox.com/get-the-deets-the-xbox-ambassadors-experience/>>. Citado na página 23.
- MICROSOFT CORPORATION. *Microsoft Visual Studio*. 2019. Disponível em: <<https://visualstudio.microsoft.com/pt-br/vs/>>. Citado na página 32.
- MICROSOFT CORPORATION. *O que é um aplicativo da Plataforma Universal do Windows (UWP)?* 2019. Disponível em: <<https://docs.microsoft.com/pt-br/windows/uwp/get-started/universal-application-platform-guide>>. Citado na página 32.
- MINCER, J. On-the-job training: Costs, returns, and some implications. *Journal of political Economy*, The University of Chicago Press, v. 70, n. 5, Part 2, 1962. Citado na página 17.
- MONTEIRO, F. *Learning single-page web application development*. [S.l.]: Packt Publishing Ltd, 2014. Citado na página 30.
- MUNIZ, A. M.; O'GUINN, T. C. Brand community. *Journal of consumer research*, Oxford University Press, v. 27, n. 4, 2001. Citado na página 20.
- OUELLETTE, E. T. N. R. C. October 1958: Physicist invents first video game. *APS News*, v. 17, n. 9, oct 2008. Citado na página 16.
- PRESSMAN, R.; MAXIM, B. *Engenharia de Software-8ª Edição*. [S.l.]: McGraw Hill Brasil, 2016. Citado 4 vezes nas páginas 35, 39, 40 e 48.
- RECLAMEAQUI. *Quase 220 mil reclamações de consumidores cansados de esperar por atendimento*. 2019. Acesso em: 05 dez. 2019. Disponível em: <https://noticias.reclameaqui.com.br/noticias/quase-220-mil-reclamacoes-de-consumidores-cansados-de-espera_3817/>. Citado na página 18.
- REILLIER, L. C.; REILLIER, B. *Platform strategy: How to unlock the power of communities and networks to grow your business*. [S.l.]: Routledge, 2017. Citado 2 vezes nas páginas 26 e 29.
- SMITH, J. Patterns - wpf apps with the model-view-viewmodel design pattern. *MSDN Magazine Issues*, v. 24, n. 2, 2009. Citado na página 35.

SMITH, S. *Overview of ASP.NET Core MVC*. 2019. Acesso em: 02 dez. 2019. Disponível em: <<https://docs.microsoft.com/en-us/aspnet/core/mvc/overview?view=aspnetcore-3.1>>. Citado na página 39.

SOMMERVILLE, I. Engenharia de software, 8 edição. *Pearson, Addison Wesley*, v. 8, n. 9, 2007. Citado 3 vezes nas páginas 32, 34 e 38.

SUBRAMANIAM, P. et al. *System and method to implement an integrated search center supporting a full-text search and query on a database*. [S.l.]: Google Patents, 2004. US Patent 6,728,702. Citado na página 48.