

UNIVERSIDADE FEDERAL DO MARANHÃO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

GENILSON SOARES BRAGA

**PROPOSTA DE API GAMIFICADA EM APLICAÇÕES WEB PARA
MOTIVAÇÃO E ENGAJAMENTO DOS USUÁRIOS FINAIS**

São Luís - MA

2013

GENILSON SOARES BRAGA

**PROPOSTA DE API GAMIFICADA EM APLICAÇÕES WEB PARA
MOTIVAÇÃO E ENGAJAMENTO DOS USUÁRIOS FINAIS**

Monografia apresentada ao Curso de Ciência da Computação da Universidade Federal do Maranhão, como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Prof. Me. Rômulo Martins França

São Luís - MA

2013

Braga, Genilson Soares.

Proposta de API gamificada em aplicações WEB para motivação e engajamento dos usuários finais/ Genilson Soares Braga. – São Luís, 2013.

97 f.

Impresso por computador (fotocópia).

Orientador: Romulo Martins França.

Monografia (Graduação) – Universidade Federal do Maranhão, Curso de Ciência da Computação, 2013.

1. Gamificação. 2. API. 3. Modelagem de software. I. Título.

CDU 004.031.4

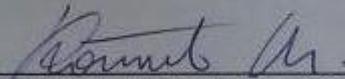
GENILSON SOARES BRAGA

**PROPOSTA DE API GAMIFICADA EM APLICAÇÕES WEB PARA
MOTIVAÇÃO E ENGAJAMENTO DOS USUÁRIOS FINAIS**

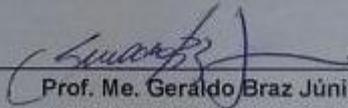
Monografia apresentada ao Curso de Ciência da Computação da Universidade Federal do Maranhão, como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação.

Aprovada em: 13 / 12 / 2013

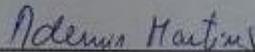
BANCA EXAMINADORA



Prof. Me. Romulo Martins França (Orientador)
Mestre em Engenharia de Eletricidade com área de concentração em Ciência da Computação
Universidade Federal do Maranhão



Prof. Me. Geraldo Braz Júnior
Mestre em Engenharia de Eletricidade com área de concentração em Ciência da Computação
Universidade Federal do Maranhão



Prof. Dr. Ademir da Rosa Martins
Doutor em Informática na Educação
Universidade Federal do Maranhão

Aos meus pais,
Raimundo Bernardo Almeida Rodrigues e
Maria de Fátima Soares Braga

AGRADECIMENTOS

A Deus em primeiro lugar, por todas as bênçãos que ele têm me proporcionado e pelos obstáculos superados.

Aos meus pais, pelo incentivo, apoio, dedicação e por sempre estarem ao meu lado me dando o apoio e a força necessária para persistir nos meus objetivos.

Aos meus irmãos, Rosana pela sua dedicação que muita das vezes me permitiu continuar sonhando, por sempre acreditar no meu esforço, Cláudio e José que em momentos difíceis estiveram ao meu lado, dando força incentivo e muitas das vezes me distraíndo, por Regina que sempre deu muito apoio.

A minha avó Florência, por ser um exemplo de vida, superação e ser uma pessoa tão maravilhosa e adorável.

A todos meus tios e tias, primos e primas, que sempre torceram por mim e me deram muita força nessa longa jornada.

Aos meus queridos amigos e amigas, Sandra, Luciana, Adriana, Rafael, Ruan, Antônio, Sanny, e Tiago, pelo incentivo, por estarem sempre ao meu lado durante todos esses anos, mesmo distantes, mas ao mesmo tempo presentes.

Aos meus tios Rafael, Eugina e seus filhos Paulo e Ana Paula, por me acolherem em sua residência, serem uma segunda família e acreditarem no meu potencial.

Aos meus colegas de curso, em especial, André, Alessandro, Caleno, Ivo, Keila, Luís Filipe, Leonardo, Rubem, Wagner e Welsson pela amizade, ajuda e esforços durante essa longa caminhada.

A todos os professores do curso e ao professor Rômulo pela orientação, paciência, ensinamentos e por sua dedicação durante este processo.

A todas as pessoas que de alguma forma me ajudaram a chegar até aqui, que sempre torceram e acreditaram na minha vitória.

“Que os vossos esforços desafiem as impossibilidades,
lembrai-vos de que as grandes coisas do homem
foram conquistadas do que parecia impossível.”

(Charles Chaplin)

RESUMO

Os perfis dos usuários que se encontram hoje na Web evoluíram muito nos últimos anos. O acesso as informações e à Internet quebrou muitos paradigmas. Uma nova perspectiva que trata a questão dos novos perfis de usuários na Web e encontra-se em ascensão é a Gamificação. Ela incorpora mecânicas e elementos de jogos em aplicações e softwares de diversas áreas. Uma das formas de se criar um serviço na Web e disponibilizá-la para a integração de diversas aplicações é a chamada Interface de Programação de Aplicações (API), em que tornou-se importante para o acoplamento de aplicações e sistemas na Web distintos, compartilhando os seus recursos disponíveis através de chamadas às interfaces. Portanto, o objetivo geral deste trabalho é propor a modelagem de uma solução de API, utilizando conceitos de Gamificação, com a finalidade de incrementar a motivação e o engajamento dos usuários finais em aplicações e serviços na Web de terceiros ou parceiros. A modelagem de Software visa proporcionar um melhor processo de desenvolvimento, amenizando a taxa de erros e aprimorando o fluxo de trabalho da equipe envolvida no desenvolvimento do produto final. A solução da API proposta é composta por dois módulos. O módulo Administrador que visa o gerenciamento e configuração da estrutura de gamificação nas aplicações dos clientes e a API gamificada que disponibiliza diversas funcionalidades para o engajamento dos usuários nestas aplicações clientes cadastradas.

Palavras-chave: Gamificação, API, Modelagem de Software.

.ABSTRACT

The profiles of the users who are on the Web today have evolved considerably in recent years. The access to information and the Internet broke many paradigms. A new perspective which addresses issues of new user profiles on the Web and is on the rise is called Gamification. It incorporates mechanical and gaming features in applications and software from various areas. One of the ways to create a web service and make it available for the integration of diverse applications is called the Application Programming Interface (API), which became important for the coupling of different Web applications and systems, sharing their resources available through calls to the interfaces. So overall aim of this work is to propose a solution for modeling API, using concepts of Gamification, in order to increase the motivation and engagement of end users in applications and Web services from third parties or partners. The modeling software seeks to provide a better development process, softening the error rate and improving the workflow of the team involved in the development of the final product. The solution proposed API consists of two modules. The Web Administrator module through the management and configuration of the structure of gamification in customer applications and API that provides several features to engage users in these applications registered customers.

Keywords: Gamification, API, Modeling Software.

LISTA DE FIGURAS E ILUSTRAÇÃO

Figura 1	Ciclo do Gamification.....	20
Figura 2	Relacionamento entre tipos de jogadores.....	22
Figura 3	Crescimento acelerado das APIs públicas.....	31
Figura 4	Relacionamento de associação entre o ator e o caso de uso.....	36
Figura 5	Relacionamento de generalização entre dois casos de uso.....	36
Figura 6	Relacionamento de inclusão entre dois casos de uso.....	37
Figura 7	Relacionamento de extensão entre dois casos de uso.....	37
Figura 8	Nomenclatura de uma classe no Diagrama de Classes.....	38
Figura 9	Relacionamento de dependência no diagrama de classes.....	38
Figura 10	Relacionamento de associação no diagrama de classes.....	39
Figura 11	Relacionamento de generalização no diagrama de classes.....	39
Figura 12	Relacionamento de agregação no diagrama de classes.....	39
Figura 13	Relacionamento de composição no diagrama de classes.....	40
Figura 14	Representação de uma ação no diagrama de atividades.....	41
Figura 15	Representação de uma ação que recebeu um evento no diagrama de atividades.....	41
Figura 16	Representação de uma ação que recebe um evento temporal no diagrama de atividades.....	41
Figura 17	Representação de um nó inicial no diagrama de atividades.....	41
Figura 18	Representação de um nó final de atividade no diagrama de atividades.....	42
Figura 19	Representação de um nó final de fluxo no diagrama de atividades.....	42
Figura 20	Representação do nó <i>fork</i> no diagrama de atividades.....	42
Figura 21	Representação de um nó de decisão no diagrama de atividades.....	43
Figura 22	Representação de um nó de objeto no diagrama de atividades.....	43
Figura 23	Representação de um nó de objeto que é sinal no diagrama de atividades.....	43
Figura 24	Interface de pontuação e fases do Duolingo.....	49
Figura 25	Interface com desempenho, emblemas e ranking do usuário no Duolingo.....	50
Figura 26	Interface de lições do Duolingo.....	50
Figura 27	Diagrama de Classes	52
Figura 28	Caso de uso principal módulo Administrador	56
Figura 29	Caso de uso Gerenciar Usuário.....	57
Figura 30	Caso de uso Gerenciar Aplicação.....	57
Figura 31	Caso de uso Gerenciar Nível.....	58
Figura 32	Caso de uso Gerenciar Missão	59
Figura 33	Caso de uso Gerenciar Recompensa.....	59
Figura 34	Caso de uso Gerenciar Meta	60

Figura 35	Diagrama de Atividades Módulo Administrador	75
Figura 36	Casos de usos da API Gamificada	76
Figura 37	Diagrama de Atividades do Módulo API Gamificada.....	84
Figura 38	Diagrama de Entidade e Relacionamento.....	87

LISTA DE ABREVIATURAS E SIGLAS

API	<i>Application Program Interface</i>
URL	<i>Universal Resource Locator</i>
UML	<i>Unified Modeling Language</i>
OMT	<i>Object Modeling Technique</i>
OOSE	<i>Object-oriented Software Engineering</i>

SUMÁRIO

1 INTRODUÇÃO	15
1.1 Motivação	16
1.2 Objetivos	17
1.2.1 Objetivo Geral.....	17
1.2.1 Objetivo Específicos.....	17
1.3 Organização do trabalho	17
2 GAMIFICAÇÃO	19
2.1 Conceitos básicos	19
2.2 Motivação em Jogos	20
2.2.1 Motivação Intrínsecas.....	20
2.2.2 Motivações Extrínsecas.....	21
2.3 Perfil dos Jogadores	22
2.4 Mecânica de Jogos	23
2.4.1 Pontos.....	24
2.4.2 Níveis.....	24
2.4.3 Ranking.....	24
2.4.4 Emblemas.....	25
2.4.5 Desafios e Missões.....	25
2.4.6 Integração.....	26
2.4.7 Ciclos de engajamento.....	26
Erro! Fonte de referência não encontrada. API – INTERFACE DE PROGRAMAÇÃO DE APLICAÇÕES	27
Erro! Fonte de referência não encontrada. Definição de API	27
3.2 Características da API	28
3.3 Tipo de APIs	30
3.4 Justificativa do Crescimento das APIs	32
4 METODOLOGIA DE MODELAGEM	33

Erro! Fonte de referência não encontrada. Especificação de

Requisitos	33
4.1.1 Requisitos Funcionais.....	33
4.1.2 Requisitos não funcionais.....	34
	Err
	o!
	Ind
	ica
	dor
	nã
	o
	def
	ini
	do.
4.2 Linguagem de Modelagem UML	
4.2.1 Diagrama de Caso de Uso.....	35
Erro! Fonte de referência não encontrada.2 Diagrama de	
Classes	37
4.2.3 Diagrama de Atividades.....	40
4.3 Modelagem Relacional	43
4.3.1 Conversão de diagrama de classes em modelo relacional.....	44
4.4 Dicionário de Dados	46
5 MODELAGEM DA API GAMIFICADA	48
5.1 Contextualização e Coleta de Dados	48
5.2 Diagrama de Classe dos Módulos	51
5.3 Módulo Web do Administrador	55
5.3.1 Diagrama de Casos de Uso.....	55
5.3.2 Requisitos Funcionais.....	60
5.3.2.1 Fluxo de evento: caso de uso Fazer Login.....	60
5.3.2.2 Fluxo de evento: caso de uso Inserir Usuário.....	61
5.3.2.3 Fluxo de evento: para o caso de uso Alterar Usuário.....	61
5.3.2.4 Fluxo de evento: para o caso de uso Excluir Usuário.....	62
5.3.2.6 Fluxo de evento: caso de uso Inserir Aplicação.....	62
5.3.2.7 Fluxo de evento: caso de uso Alterar Aplicação.....	63

5.3.2.8 Fluxo de evento: caso de uso Excluir Aplicação.....	64
5.3.2.9 Fluxo de evento: caso de uso Consultar Aplicação.....	64
5.3.2.10 Fluxo de evento: caso de uso Inserir Nível.....	65
5.3.2.11 Fluxo de evento: caso de uso Alterar Nível.....	65
5.3.2.12 Fluxo de evento: caso de uso Excluir Nível.....	66
5.3.2.13 Fluxo de evento: caso de uso Consultar Nível.....	66
5.3.2.14 Fluxo de evento: caso de uso Inserir Missão.....	67
5.3.2.15 Fluxo de evento: caso de uso Alterar Missão.....	67
5.3.2.16 Fluxo de evento: caso de uso Excluir Missão.....	68
5.3.2.17 Fluxo de evento: caso de uso Consultar Missão.....	69
5.3.2.18 Fluxo de evento: caso de uso Inserir Meta.....	69
5.3.2.19 Fluxo de evento: caso de uso Alterar Meta.....	70
5.3.2.20 Fluxo de evento: caso de uso Excluir Meta.....	70
5.3.2.21 Fluxo de evento: caso de uso Consultar Meta.....	71
5.3.2.22 Fluxo de evento: caso de uso Inserir Recompensa.....	71
5.3.2.23 Fluxo de evento: caso de uso Alterar Recompensa.....	72
5.3.2.24 Fluxo de evento: caso de uso Excluir Recompensa.....	72
5.3.2.25 Fluxo de evento: caso de uso Consultar Recompensa.....	73
5.3.3 Requisitos Não Funcionais.....	73
5.3.3.1 Usabilidade.....	74
5.3.3.2 Documentação do Usuário.....	74
5.3.3.3 Comunicação.....	74
5.3.3.4 Segurança.....	74
5.3.3.5 Disponibilidade.....	74
5.3.4 Diagrama de Atividades.....	74
5.4 Módulo da API Gamificada.....	75
5.4.1 Casos de Uso.....	75
5.4.2 Requisitos Funcionais.....	76
5.4.2.1 Fluxo de evento: caso de uso Recuperar Ranking.....	76
5.4.2.2 Fluxo de evento: caso de uso Recuperar Pontuação Usuário.....	77
5.4.2.3 Fluxo de evento: caso de uso Recuperar Colocação Usuário.....	78
5.4.2.4 Fluxo de evento: caso de uso Recuperar Recompensa Usuário.....	78

5.4.2.5 Fluxo de evento: caso de uso Recuperar Meta Usuário.....	79
5.4.2.6 Fluxo de evento: caso de uso Recuperar Missão Usuário.....	80
5.4.2.7 Fluxo de evento: caso de uso Recuperar Nível Usuário.....	80
5.4.2.8 Fluxo de evento: caso de uso Registrar Recompensa Usuário.....	81
5.4.2.9 Fluxo de evento: caso de uso Registrar Meta Usuário.....	82
5.4.3 Requisitos Não Funcionais.....	83
5.4.3.1 Disponibilidade.....	83
5.4.3.2 Manutenção.....	83
5.4.3.3 Segurança.....	83
5.4.3.4 Desempenho.....	83
5.4.4 Diagrama de Atividades.....	83
5.5 Diagrama de Entidade Relacionamento.....	84
5.5.1 Dicionário de Dados.....	87
6 CONCLUSÃO.....	88
REFERÊNCIAS.....	89
APÊNDICE I – DICIONÁRIO DE DADOS.....	91

1 INTRODUÇÃO

O evidente aumento de usuários na rede mundial de computadores, o compartilhamento e a colaboração de conteúdos estão aumentando de maneira significativa. Com isto, muitas aplicações, recursos e funcionalidades foram criadas na Web para suprir novas necessidades e em larga escala, resultando muitas vezes na falta de motivação e tempo disponível dos usuários para consumirem parte do que está disponível.

A geração dos usuários evoluiu nos últimos anos sendo classificadas em três diferentes abordagens marcadas pelo período de nascimento. A “Geração X” é constituída por indivíduos que nasceram entre 1966 a 1975, a “Geração Y” entre o final da década de 70 e o início dos anos 90 e após este período são considerados da “Geração Z”.

Para diferenciar estas gerações, pode-se citar o acesso às informações e à Internet. Os indivíduos da Geração X são empreendedores, pragmáticos, voltados para ação, impõem grandes resistências à mudanças e são conservadores dentro da organização. Já os da Geração Y, por exemplo, nasceram durante o surgimento da Internet, em sua maioria são profissionais que possuem o raciocínio rápido e, conseqüentemente, ansiosos quanto aos resultados e projetos.

Já na Geração Z encontram-se os novos indivíduos que estão adentrando no mercado de trabalho, não se retêm a fazer apenas uma atividade por vez e não concebem o mundo sem computadores, celulares e Internet. Para eles o conceito de globalização é natural e espontâneo, pois o relacionamento pessoal é facilmente substituído pelo relacionamento virtual. São o que se pode chamar de “*net generation*” ou nativos digitais, segundo Prensky (2007).

Diante de tantas tecnologias e serviços disponíveis na Web como as redes sociais, sistemas colaborativos, jogos e dispositivos móveis, manter a atenção e motivação dos usuários em uma aplicação Web não é uma tarefa fácil. Portanto, existe a necessidade de se implementar técnicas ou estratégias em aplicações e serviços na Web que possibilitem a motivação e engajamento dos usuários, motivando-os em sua participação e colaboração com os demais membros de uma rede ou nos ambientes das aplicações.

Uma nova perspectiva que trata esta questão e que incorpora mecânicas de jogos à aplicações e softwares de diversas áreas, encontra-se em ascensão, sendo

conhecida como Gamificação, neologismo oriundo do inglês *Gamification*. Refere-se ao uso de elementos de jogos digitais em contextos não associados a jogos, utilizado em produtos e serviços para melhorar a experiência de seus usuários, conforme Deterding et al. (2011).

Muitas aplicações e serviços na Web estão sendo integrados para facilitar o seu acesso e uso, pois existe uma diversidade de linguagens e plataformas que necessitam de compartilhamento de informações, funcionalidades e recursos.

Uma das formas para à integração supracitada é a chamada API – Interface de Programação de Aplicações (do inglês, *Application Program Interface*), no qual é importante para o acoplamento de aplicações e sistemas na Web distintos em que se possam compartilhar os seus recursos disponíveis através de chamadas às interfaces.

Às APIs são usadas por programas ou subsistemas existentes e oferecem uma série de serviços que são acessados por chamadas de procedimentos de interface, de acordo com Sommerville (2011).

1.1 Motivação

Tendo uma visão mais ampla, pode-se apontar vários motivos pelo qual a Gamificação é uma solução adequada ao perfil de muitos usuários em aplicações na Web. Segundo Stanley (2012), o envolvimento tecnológico aliado ao conceito de Gamificação, estimula o usuário a praticar a interação, diálogo e simulação de situações reais de comunicação.

Os indivíduos da geração atual possuem um conjunto de características que se enquadram no perfil de usuário de jogos eletrônicos, Van Eck (2006). As aplicações e serviços disponíveis na Web juntamente com os perfis de usuários da referida geração, que tendem a ser colaborativos e participativos, é propício para utilização de ambientes gamificados.

Percebe-se que a geração atual de usuários da Web possui elevada afinidade com os jogos eletrônicos e suas características como a objetividade e respeito às regras claras, percepção imediata de objetivos atingidos, alto nível de interatividade, satisfação na solução de desafios e existência de um nível de competição. Sendo que isso é associado a um elevado nível de envolvimento e motivação dos usuários e são vantagens apontadas para o uso dos jogos eletrônicos, segundo Kickmeier-Rust (2007).

Portanto, este trabalho justifica-se por envolver uma proposta que sistematize a modelagem de uma API de fácil uso e integração, com os conceitos de Gamificação, para o incremento do engajamento e motivação dos usuários em aplicações e serviços na Web de terceiros ou parceiros. Além disso, o trabalho visa nortear futuras implementações e integrações sobre Gamificação em aplicações Web por desenvolvedores interessados no assunto.

1.2 Objetivos

1.2.1 Objetivo Geral

Propôr a modelagem de uma solução de API, utilizando conceitos de Gamificação, com a finalidade de incrementar a motivação e o engajamento dos usuários finais em aplicações e serviços na Web de terceiros ou parceiros.

1.2.2 Objetivos Específicos

- a) Levantar os requisitos em aplicações Gamificadas, literatura e APIs disponíveis na Web para desenvolvimento da API proposta;
- b) Analisar os requisitos levantados para desenvolvimento da solução da API Gamificada;
- c) Documentar os requisitos de desenvolvimento da solução da API Gamificada para aplicações e serviços na Web.
- d) Elaborar os diagramas de casos de uso, diagrama de classe, diagrama de atividades e diagrama de entidade e relacionamento, seguindo a UML para a solução da API Gamificada.

1.3 Organização do trabalho

O trabalho encontra-se dividido em 6 capítulos. No Capítulo 2 é realizado um levantamento teórico sobre Gamificação e seus conceitos básicos, abordando como é a motivação e mecânicas dos jogos, além do perfil dos jogadores.

O Capítulo 3 trata do referencial teórico sobre API, apresentando os seus conceitos básicos e definições, a sua importância no mundo atual como soluções tecnológicas e de negócios para suprirem as novas necessidades das pessoas e empresas e o seu grande crescimento na Web.

O Capítulo 4 apresenta os conceitos básicos e teóricos sobre a modelagem de software abordando os requisitos, a linguagem de modelagem UML e os seus diagramas, modelagem relacional e o dicionário de dados.

O Capítulo 5 apresenta a modelagem desenvolvida para a solução da API com conceitos de Gamificação, em que elaborou-se os requisitos funcionais e não funcionais, diagramas de casos de uso, diagrama de classe, diagrama de atividades, modelo relacional e dicionário de dados.

O Capítulo 6 apresenta as conclusões aferidas deste trabalho, fazendo uma retrospectiva geral. Também são avaliados se os objetivos foram alcançados e quais são os trabalhos futuros referentes ao tema abordado.

2 GAMIFICAÇÃO

Este capítulo apresenta os conceitos e as definições básicas sobre Gamificação, em que descreve-se a finalidade da sua aplicação, a motivação e mecânicas dos jogos, além do perfil dos jogadores. Neste capítulo é levado em consideração a literatura especializada sobre o assunto.

2.1 Conceitos básicos

A Gamificação tem atraído o interesse tanto da indústria quanto da academia durante os últimos anos, conforme cita Deterding et al. (2011), McGonigal (2011) e Bogost (2007). Por exemplo, o sucesso de serviços móveis como Foursquare e Nike+ são frequentemente atribuídos a Gamificação (DETERTING et al. 2011).

De acordo com Zichermann (2011), a Gamificação está relacionada à inclusão da mecânica, estilo e pensamento envolvendo técnicas de design de jogos, buscando o envolvimento de pessoas na solução de um problema. Logo de início, é importante frisar que apesar do uso de elementos de jogos, necessariamente, as aplicações gamificadas não se tornam jogos.

Na mesma linha, Kapp (2007) refere-se à Gamificação como a mecânica, estética e raciocínio correntes nos jogos, provendo formas de engajar as pessoas, conduzindo-as para soluções de problemas. Deterding et al. (2011), reforçam que a inserção de elementos de jogos a contextos não lúdicos tem o potencial de gerar, entre os participantes, um nível de envolvimento semelhante ao que é conseguido com os jogos.

Algumas aplicações que utilizam os conceitos de Gamificação estabelecem algum tipo de recompensa pelas atividades realizadas pelo usuário. Estes sistemas utilizam as mecânicas de jogos mais comuns, tais como emblemas, níveis, quadros de liderança, metas e pontos. Esta estrutura é citada por Nicholson (2012) como BLAP da Gamificação, que correspondem aos Emblemas, Quadro de Líderes, Conquistas e Pontos (do inglês, *Badges, Leaderboards, Achievements e Points*).

Seguindo a especificação de Liu, Alexandrova e Nakajima (2011), para usar a Gamificação em um ambiente não associado a jogo, pode-se criar um ciclo, de acordo com a Figura 1 abaixo. Nele, a interação inicia a partir de um objetivo ou desafio claro com uma condição de vitória específica. Ao alcançar determinada meta, algumas recompensas são dadas nesse sentido, no qual é normalmente

gerenciado por um sistema de pontos (pontuação, moeda virtual, pontos de experiência etc). Ao se tomar como base o sistema de pontos e históricos de conquistas, um quadro de líderes (globais ou parciais) e emblemas são fornecidos aos jogadores para motivar suas capacidades de competitividade e que possivelmente resultem na mudança no estado (*status*) dentro do sistema, o que possibilitam aos jogadores maior visibilidade em suas redes sociais.

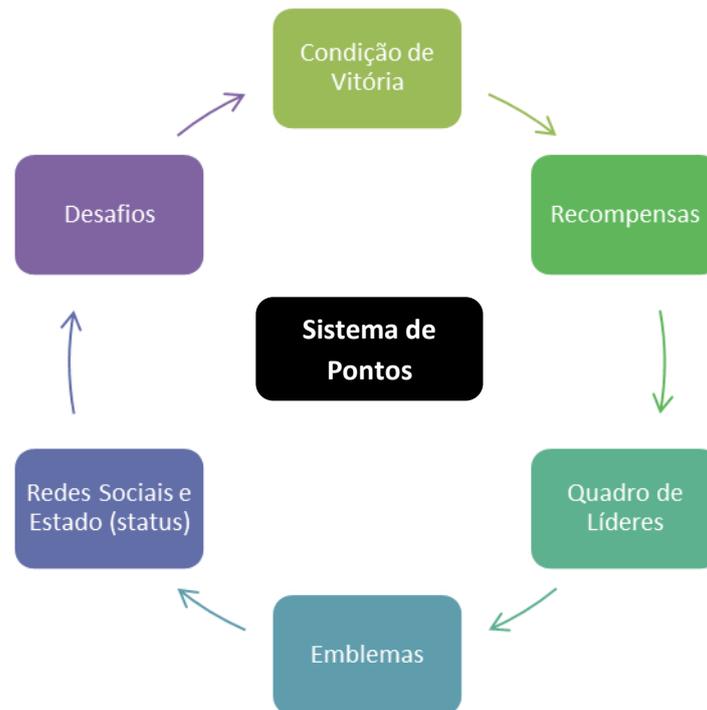


Figura 1 – Ciclo da Gamificação
 Fonte: Liu, Alexandrova e Nakajima (2011)

2.2 Motivação de Jogos

Mas quais são os tipos de motivações dos usuários que existem? Para que uma aplicação Gamificada seja bem sucedida é necessário compreender os tipos de motivações existentes, segundo Zichermann (2011), que podem-se dividir em intrínsecas e extrínsecas.

2.2.1 Motivações Intrínsecas

São oriundas de fatores internos ao indivíduo. Estão diretamente relacionadas na forma de ser do jogador, seus interesses e gostos. Para este tipo de motivação, não há muita necessidade de existir recompensas, pois a tarefa em si própria é o interesse para o usuário. Esta forma de motivação é constante, e depende

exclusivamente do sujeito e não de fatores externos. Assim a tarefa não representa uma obrigação, um meio para atingir um fim (recompensa) e, sim, para representar uma satisfação pessoal. Pode-se dizer que a motivação intrínseca está relacionada inteiramente com felicidade e realização pessoal.

Para representar o sentimento de uma pessoa, enquanto ela participa de uma aplicação gamificada, é necessário analisar o seu principal objetivo, se ele está ligado ao engajamento social e se sente prazer em jogar ou usar o ambiente. Por exemplo, realizar tarefas sem pensar nas consequências ou benefícios pode ser considerada como motivação intrínseca, em que o fator pessoal do jogador é responsável pelo tipo de conteúdo que será compartilhado com os outros. Dessa maneira, o jogador prefere participar de um processo de interação com outros usuários, do que simplesmente fazer outras ações.

Para continuar motivado, o usuário necessita que a aplicação gamificada lhe proporcione elementos para engajá-lo em seu uso. No entanto, deve-se ter cuidado para não modificar drasticamente o cenário do usuário. Pois, no momento que o jogo ou uma aplicação gamificada não produz mais as mesmas sensações que eram obtidas no início, no caso sensações de prazer para o usuário, é provável que ele abandone-o.

2.2.2 Motivações Extrínsecas

É originada por fatores externos ao indivíduo. A principal motivação do indivíduo para realizar uma tarefa é a busca por recompensa (monetária). A punição ou recompensa são motivadores importantes para este tipo de indivíduo. Baseado nesse tipo de perfil, pode-se determinar a relação do jogador com as recompensas que são disponibilizadas no jogo ou em uma aplicação. Se elas forem retiradas da aplicação, o sujeito tende a deixar de se mobilizar, de estar motivado, pois não haverão motivos suficientes para continuar a executar suas tarefas, visto que não possuirá perdas ou ganhos.

De acordo com Zichermann (2011), este tipo de motivação é muito inconstante, visto que depende exclusivamente de fatores externos. Para o indivíduo não importa a tarefa em si, mas sim a recompensa que a tarefa ao ser executada lhe trará, o que implica na ausência de satisfação e prazer apenas na execução da tarefa.

Este tipo de jogador sempre observa as recompensas, a vontade de conseguir sempre mais itens é suprida com o ganho de mais recompensas (itens, dinheiro, moedas, estalecas etc). Estes fatores externos influenciam desde a entrada de um novo usuário até sua permanência ou saída do jogo. Porém estas motivações requerem análise para sua implantação. Caso ocorra falha no processo de implantação, o usuário não se sentirá mais motivado e engajado, acarretando a sua saída da aplicação gamificada.

2.3 Perfil dos Jogadores

Ao tratar do assunto de Gamificação torna-se fundamental apresentar os perfis de jogadores existentes.

De acordo com Bartle (1990), existem basicamente quatro tipos de jogadores: Exploradores, Sociais, Conquistadores e Externos. Essas motivações que levam um usuário a jogar não está diretamente ligada a apenas um tipo de tipo de jogador. O mesmo jogador pode ter um perfil com um conjunto destes tipos. Frisa-se que estes tipos não são um padrão, são somente uma forma de classificação, buscando criar um contexto para cada tipo deles.

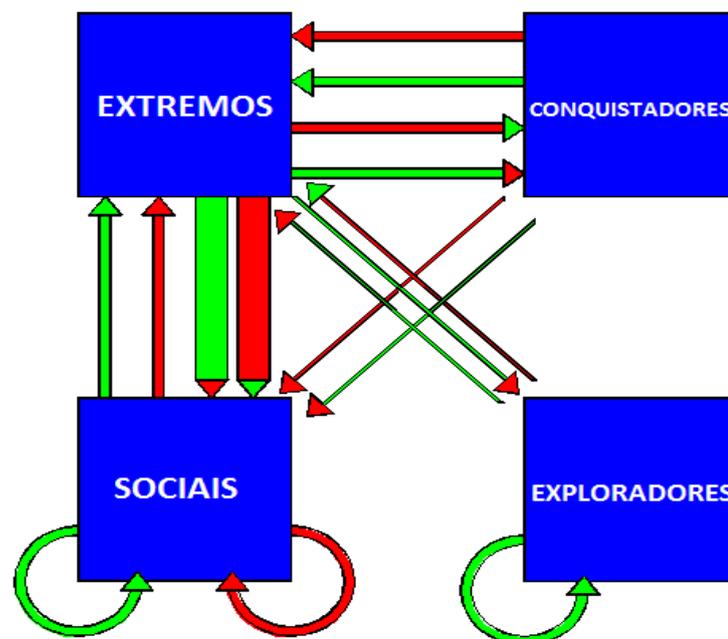


Figura 02: Os tipos de jogadores.

Fonte: Adotado de Bartle (1990)

Abaixo são apresentados os quatro tipos de jogadores, conforme a visão Bartle (1990), de acordo a Figura 02:

- **Exploradores** (*Explorers*): são conduzidos pela motivação de descobrir o máximo possível sobre o jogo, incluindo desde o mapeamento da área geográfica, de elementos visíveis ou não, até a compreensão da mecânica. Possuem muita curiosidade e vão buscar entender o porquê e como cumprir cada desafio que é proposto.
- **Sociais** (*Socialites*): são interessados em pessoas, pois prezam pela interação entre elas e no que elas têm a dizer. O jogo é apenas um contexto para socializarem com outros jogadores. A principal preocupação não é vencer o jogo, mas buscar a socialização. Eles são os maiores comentadores de status, resultando na motivação e no engajamento de usuários para desafios em equipe.
- **Conquistadores** (*Achievers*): são movidos por objetivos no jogo para se destacar dos outros jogadores, preferem jogos que possam fazer um planejamento, geralmente alguma forma de acumular pontos. Entre eles estão os pontos de experiência, níveis ou mesmo cupons de desconto. Este tipo de jogador compõe uma parte integral de qualquer jogo competitivo. Eles certamente serão atraídos por um conjunto de emblemas ou troféus, por exemplo.
- **Extremos** (*Killers*): é o perfil mais competitivo. Normalmente são movidos pela vontade de impôr-se e ficam satisfeitos em proporcionar momentos de agonia e ansiedade nos outros jogadores. Eles necessitam que os outros vejam isso, caso contrário, não seria uma legítima vitória. Para que eles se sintam vitoriosos, alguém precisa perder.

2.4 Mecânica de Jogos

Após apresentar as definições sobre Gamificação, tipos de motivações e perfis dos jogadores, aborda-se neste item à questão das mecânicas dos jogos.

De acordo com Zichermann (2011), a mecânica de um jogo é desenvolvida com base em uma diversidade de características, que se forem usadas de forma correta fornecem uma resposta satisfatória para o usuário. Estas características,

conforme o autor, são os Pontos, Níveis, Ranking, Emblemas, Desafios e Missões, Integração e Ciclos de Engajamento.

2.4.1 Pontos

Os pontos são uma das características mais importantes nos ambientes de jogos, independente de qual forma de pontuação será adotada, é necessário que o usuário sinta-se confortável com a pontuação escolhida. Ela pode ser usada para checar o nível de interação do usuário com o jogo.

Kapp (2012) cita que as principais pontuações estão sempre visíveis ao usuário de uma forma que não atrapalhe à visualização do jogo, um local em que o jogador sempre possa fazer uma verificação rápida.

2.4.2 Níveis

Os níveis tem o papel de marcar o progresso do usuário em um jogo. Eles podem ter diferentes representações, umas das principais diferenças é a troca de cenário, pois podem ser mudados os elementos que vão compor o jogo, assim como o comportamento destes elementos no jogo. O uso consciente destas alterações para cada nível deve ser utilizado de forma correta, acarretando em uma transição agradável, segundo Zichermann (2011).

Ao perceber que houve uma alteração no nível do jogo, os elementos que compõem a complexidade e os cenários do jogo, também são alterados. As formas de criação destes níveis devem ser analisadas e validadas pelos desenvolvedores, adequando uma solução específica para o ambiente.

A complexidade dos níveis não crescem de uma forma linear, ou seja, o nível mais a frente não é X vezes mais complexos que o anterior, segundo Kapp (2012). Então seria necessária uma função curvilínea de complexidade para que o jogador possa ir se habituando ao jogo, pois a alteração da dificuldade torna-se mais sutil. Já outra possibilidade para este quesito é a intercalação de dificuldades.

2.4.3 Ranking

É o mecanismo de listagem das melhores pontuações, pois define as diferentes posições de estado dos jogadores, de acordo com Kapp (2012). Existem várias formas de usar *rankings*. No início da aplicação dos conceitos dos jogos, os rankings mostravam de uma vez todos os jogadores que obtiveram as melhores

pontuações. Algumas pessoas ao visualizarem estes valores, já sentiam-se desmotivadas a ultrapassar os pontos dos outros jogadores, segundo Zichermann (2011).

O ideal é que o ranking mostre apenas o necessário ao jogador, normalmente sua posição, a colocação dos seus amigos e a posição do resultado que deve ser ultrapassado. Para proporcionar maior motivação, há necessidade que esta posição esteja no mínimo uma posição acima.

Para incrementar a motivação e o engajamento dos usuários é necessário manter as melhores pontuações no topo do ranking, assim como as diferentes formas e classificações de pontuações. Outro modo de criar rankings são os que não devem funcionar com visualizações públicas, pois os integrantes das posições deste tipo de ranking podem ficar constrangidos com o resultado e o significado dos valores.

2.4.4 Emblemas

São usados para referenciar as informações referentes a um determinado usuário ou sobre suas ações, que a levaram a ganhar um determinado emblema. Eles podem aumentar a competitividade e melhorar a interação no jogo, pois possibilita fazer que o usuário execute ações esperadas pelo ambiente. O jogador é motivado a ganhar a maior quantidade de emblemas possíveis ou apenas ser o detentor de um grande feito ou passar informações de como conseguiu o mesmo. Um mecanismo adequado de emblemas encoraja o jogador a continuar no jogo tentando conseguir mais emblemas.

2.4.5 Desafios e Missões

Desafios e missões norteiam os jogadores novatos, que normalmente não sabem o que fazer primeiro em um jogo ou aplicação gamificada. Caso o jogador fique ocioso, logo no início da criação de sua experiência, este não terá interesse em continuar. Com as missões é possível ocupar o tempo do jogador, proporcionar diversão, dar diferentes tipos de desafios (que devem corresponder, preferencialmente, ao nível do jogador) e recompensas. O propósito é sempre fornecer missões ao jogador que poderão ser realizadas sequencialmente ou permitir o usuário escolher quais fazer.

As missões que requerem a formação de grupos de outros jogadores para serem concluídas, geralmente, são muito complexas. A organização de um grupo com diferentes tipos de jogadores pode ser uma tarefa um pouco complicada, métodos devem ser aplicados para que os jogadores possam querer fazer estes tipos de missões. Os desafios em comum, recompensas e emblemas podem ser uma solução.

2.4.6 Integração

É o processo de inserir novos usuários aos jogos ou ambientes gamificados. Os primeiros minutos de experiência são importantes para eles continuem a usar o ambiente. Percebe-se a necessidade dos usuários criarem uma experiência básica do ambiente, sendo importante citar que caso o usuário não consiga se adaptar no primeiro contato, a possibilidade de retorno dele declina.

Deve ser evitado o uso exagerado de texto pra tentar introduzir conhecimento sobre o sistema para o novo usuário. Ele deseja conhecer o ambiente, mas não quer passar muito tempo lendo. Nos primeiros contatos, o ambiente deve mostrar ao jogador o que ele pode usá-lo, dando tarefas ao usuário resultando em sucesso e recompensas. Se o usuário perder no início, ele se sentirá desmotivado a prosseguir.

2.4.7 Ciclos de Engajamento

O ambiente não deve se preocupar apenas com os primeiros acessos do usuário, mas também deve fazer com que ele retorne após sua saída, ou seja, como manter o jogador envolvido.

Um dos métodos para realização desta tarefa são os ciclos de engajamento social. Pois a socialização, segundo Zichermann (2011), é o principal fator com que fará o usuário retornar ao ambiente gamificado. O usuário depois que sai pra resolver coisas pessoais, deverá possuir a vontade de saber o que aconteceu em seu afastamento e ficar na expectativa de retornar ao jogo ou ambiente gamificado.

3 API – INTERFACE DE PROGRAMAÇÃO DE APLICAÇÕES

Neste capítulo são apresentados os conceitos básicos sobre API – Interface de Programação da Aplicações.

3.1 Definições de API

Os softwares geralmente tem uma interface de usuário, na qual as pessoas entram com dados e recebem informações do sistema. Quando um software precisa se comunicar com outro, em vez de se comunicar com humanos, eles fazem isto através de uma interface diferente, customizada para este tipo de comunicação, a chamada API.

API é uma forma de comunicação entre os computadores através de uma rede, geralmente a Internet, utilizando uma linguagem comum em que ambos compreendem, de acordo com Jacobson et al. (2012).

Blanchette (2008) diz que a API é um conjunto de símbolos que são exportados e disponíveis para os usuários de uma biblioteca para desenvolver suas aplicações. Na prática, uma API significa ou indica uma interface bem definida que especifica e fornece um conjunto de serviços à outras aplicações de desenvolvedores terceiros ou parceiros.

Uma API pode fornecer um caminho para desenvolvedores terceiros no acesso aos dados e serviços para construção de aplicações. As APIs do Twitter e Facebook são exemplos de uso. Existem APIs que são abertas para qualquer desenvolvedor, outras apenas para sócios e algumas que são usadas internamente, uso restrito, para ajudar a administrar melhor um negócio e facilitar a colaboração entre equipes.

Jacobson et al. (2012) complementa que uma API é essencialmente um contrato. Uma vez que este contrato esteja em vigor, os desenvolvedores são atraídos para usá-la, porque sabem que podem consumir os seus serviços. O contrato aumenta a confiança, o que aumenta o uso. O contrato também faz a ligação entre o provedor da API e os seus consumidores, aplicações de desenvolvedores terceiros ou parceiros, de forma mais eficiente, pois as interfaces tendem a ser bem documentadas, consistentes e previsíveis.

A API é fornecida por um provedor e quem as usam são as aplicações de desenvolvedores parceiros ou terceiros. As pessoas que usam as estas aplicações

são os chamadas de usuários finais, sendo considerado o público secundário da API e, muitas vezes, o público que norteia futuras mudanças conforme as necessidades de negócio.

Para exemplificar e compreender mais sobre o conceito de API, pode-se compará-la a um website. Um website geralmente fornece informações sobre demanda. Uma empresa insere o seu conteúdo na Web e as pessoas consomem. Eles não têm contratos ou estruturas em torno da utilização de conteúdo. Se o seu conteúdo for alterado, os visitantes que o acessam obtém esse novo conteúdo sem problemas. Os navegadores (*browsers*) não são afetados e qualquer mudança é transparente ao usuário. Ainda se o website for reformulado drasticamente, o único impacto é sobre o usuário acostumado a ver o conteúdo, no qual irá se adaptar ao novo design e encontrar a informação ou serviço que necessita.

Já uma API é completamente diferente, porque ela tem um contrato e as aplicações são desenvolvidas seguindo as regras deste contrato. No parágrafo anterior tem-se o exemplo de uma interação entre website e humano, agora a interação é entre API e uma aplicação de terceiro ou parceiro. Programas, ao contrário dos humanos, não são flexíveis. Se os fornecedores da API alterarem qualquer regra no contrato, o efeito cascata de incompatibilidades sobre as aplicações de terceiros desenvolvidas pode ser potencialmente grande.

3.2 Características da API

De acordo com Jacobson et al. (2012), uma API deve seguir as especificações conforme abaixo:

- O provedor da API deve descrever exatamente quais funcionais oferecerá;
- O provedor da API deve descrever quando uma funcionalidade estará disponível e quando ela deve mudar para evitar incompatibilidades;
- O provedor da API deve esboçar limitações técnicas adicionais dentro da API, tais como limites de frequência que controlam quantas vezes uma aplicação ou usuário final é permitido usar a API em uma determinada hora, dia ou mês;
- O provedor da API deve prover uma descrição das restrições legais ou de negócios (contrato) ao usar a API;

- O aceite de uso da API deve ser acordado pelos desenvolvedores das aplicações para usá-la somente como descrito e seguindo as regras estabelecidas pelo provedor da API;

Além disso, os provedores da API devem oferecer ferramentas aos consumidores como:

- Mecanismos para acessar a API e entender os seus termos de uso;
- Documentação para auxiliar na compreensão da API;
- Recursos como exemplos de programas e comunidades de desenvolvedores para oferecer suporte à aqueles que utilizam a API;
- Informações operacionais sobre o estado da API e à questão da sua demanda de uso;

Blanchette (2008) traz a sua visão a partir de leituras e experiências sobre o que uma API deve ser como de fácil aprendizado e memorização, código legível, difícil de usá-la indevidamente, fácil de estender e completa.

- Fácil aprendizado e memorização:** é quando apresenta convenções de nomenclaturas consistentes e padrões, economia de conceitos e previsibilidade. Uma API deve ser consistente para ser de fácil memorização, pois há pouco para se lembrar. Sendo que ela não é composta apenas pelos nomes das classes e métodos, mas também pela sua semântica pretendida. A semântica deve ser simples e clara. Algumas APIs são de difícil entendimento, pois exigem que os usuários escrevam alguns códigos desnecessários, que deveriam ser abstratos, só para começar a usá-la. Um exemplo de uma API de fácil aprendizado possibilita escrever o exemplo do "Hello World" em apenas algumas linhas de códigos e que permite gradativamente expandi-lo para obter as aplicações mais complexas.
- Código Legível:** está relacionado com a facilidade para documentar e manter a aplicação. Também é menos provável que contenha erros, porque eles são mais visíveis com a legibilidade do código.
- Difícil de usar de forma indevida:** uma API bem projetada torna mais fácil a escrita de código correto do que incorreto, além de incentivar as boas práticas de programação.
- Fácil de estender:** as bibliotecas (conjunto de subprogramas e rotinas) da API se expandem ao longo do tempo. Novas classes surgem e as já

existentes obtêm novos métodos, que adiciona novos parâmetros e assim por diante. As APIs devem ser projetadas com este pensamento. Quanto maior a API, maior são as dúvidas e confrontos entre os novos conceitos e os já existentes nas possíveis expansões.

- e) **Completa:** deve ser abrangente em áreas abordadas do negócio tratado na API e deve permitir que os usuários realizem diversas chamadas. Raramente será possível fornecer uma API completa para uma área, mas pelo menos deve ser possível que os usuários possam realizar a sua extensão e personalizá-la criando subclasses.

A API deve ser tratada como um produto de software, tendo em conta o controle de versão, compatibilidade e tempo de evolução para acomodar qualquer nova funcionalidade que venha ser implementada. Deve haver um equilíbrio entre o apoio da sua base existente e, ao mesmo tempo, a manutenção das alterações necessárias para que a API possa crescer de acordo com o negócio e seguindo uma evolução planejada.

Jacobson et al. (2012) ainda cita que, uma API que encontra-se na Web é um produto on-line que pode mudar quase constantemente para atender às necessidades do negócio ou para servir a carga de tráfego atual da forma mais eficiente. Mas essas são mudanças da implementação (*backend*) e não da interface. Quando mudanças são bem realizadas, a implementação de uma API pode mudar diariamente ou mesmo mais vezes, enquanto a interface continua a ser consistente.

3.3 Tipos das APIs

Segundo Jacobson et al. (2012) existem dois tipos de APIs: públicas e privadas. Atualmente as APIs privadas são mais prevalentes como por exemplo o Facebook e Twitter.

O crescimento das atividades relacionadas as APIs podem ser monitoradas observando as APIs públicas, já que as APIs privadas são de difícil controle em função dos seus proprietários. O serviço na Web ProgrammableWeb.com acompanha a criação das APIs de acesso público. Abaixo, conforme Jacobson et al. (2012), pode ser visualizado o crescimento acelerado destas APIs na Figura 3.

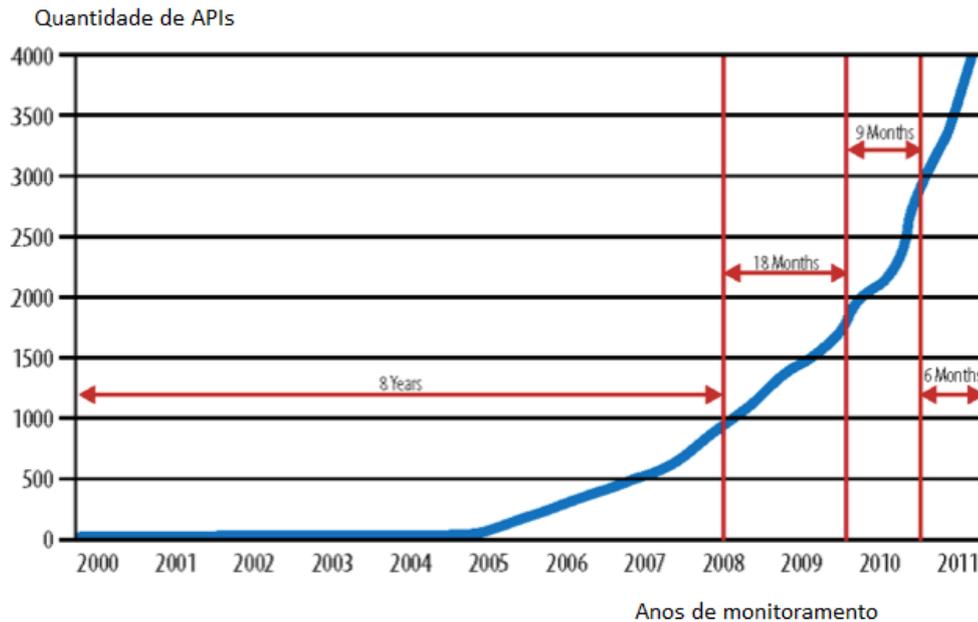


Figura 3: Crescimento acelerado das APIs públicas

Fonte: www.programmableweb.com

A Figura 3 acima representa apenas uma fração das APIs, mas permite ter uma visão sobre o processo. O referido website não acompanha apenas APIs públicas, mas segundo a experiência do autor, indica que APIs privadas estão desfrutando similarmente de um super crescimento. Além disso, acredita-se que as APIs privadas já são substancialmente mais importantes para a maioria das empresas de APIs públicas.

Alguns dados sobre os serviços aos consumidores e negócios populares mostra como as APIs se tornaram o principal veículo de tráfego na Web. Websites como Twitter, Google, Netflix, eBay, Salesforce e outros geram mais da metade de seus tráfegos através de APIs. Abaixo pode ser observado as seguintes estatísticas, segundo Jacobson et al. (2012),:

- Twitter: mais de 15 milhões de chamadas por dia a partir de julho de 2011, com 75% do tráfego que é obtido através da API;
- Netflix: mais de 1 bilhão de chamadas por dia em outubro de 2011;
- Amazon Web Services: mais de 260 bilhões de objetos armazenados no Amazon S3 – Serviço de Armazenamento Simples (do inglês, *Simple Storage Service*), a partir de janeiro 2011;
- NPR - *National Public Radio*: 3,2 bilhões de histórias entregues através da API por mês em outubro de 2011;
- Google: 5 bilhões de chamadas por dia;

- Facebook: 5 bilhões de chamadas por dia;

Em face dos dados apresentados, torna-se evidente que as APIs estão se tornando um canal de volume imenso para comunicação, comércio e negócios de todos os tipos.

3.4 Justificativa do Crescimento das APIs

Como as APIs alcançaram um relevante tráfego de dados na Web, conforme já apresentados no tópico anterior? Ressaltam-se três motivos que descrevem como as APIs atingiram o momento atual de importância em soluções na Web, segundo Jacobson et al. (2012):

- 1) Maturidade do processo:** as APIs não tratam apenas de tecnologia, como em muitos problemas de negócios, o que realmente se tem são os problemas ou necessidades de pessoas ou empresas. Elas evoluíram oferecendo um padrão comum para resolução destas necessidades de forma colaborativa.
- 2) Auto-serviço:** por que o código aberto (*open source*) fez sucesso? Embora a disponibilidade do código fonte é, muitas vezes, o foco das discussões sobre o sucesso do código aberto, a ideia de auto-serviço se tornou relevante. Em muitas pesquisas foram comprovadas que apenas uma pequena percentual de desenvolvedores queriam ler ou modificar o código-fonte. Os provedores de APIs aprenderam com o código aberto por: a) disponibilizarem muitas APIs sem cobrar nada; b) estarem mantidas em uma base de auto-serviço; c) serem de fácil uso; d) terem comunidades online ativas, onde os membros mais ativos não trabalham os provedores da API, e sim, ajudam porque a API é fundamental para o que eles fazem e se sentem realizados a ajudar os outros a verem o seu valor.
- 3) Maturidade tecnológica:** apesar dos profissionais da computação e informática terem usados as APIs por décadas, poucas pessoas percebem que o ápice das atividades na Web estão acontecendo através delas, com o Twitter, Netflix e outros serviços on-line já citados. O resultado final, que pode ser percebido, é uma grande quantidade de tráfego para mudar os negócios de milhares de pessoas e empresas.

4 METODOLOGIA DE MODELAGEM

Neste capítulo são apresentados os conceitos básicos sobre Modelagem de Software.

4.1 Especificação de Requisitos

Ao se descrever as funções e as restrições de um sistema, definem-se os requisitos. “A Engenharia de Requisitos tem o objetivo de descobrir, analisar, documentar e verificar essas funções e restrições” (SOMMERVILLE, 2011).

De acordo com Pfleeger (2004), esta etapa é considerada uma das essenciais no ciclo de desenvolvimento do software, caso esta especificação seja feita de maneira equivocada, incompleta ou inconsistente, pode-se acarretar em um desenvolvimento problemático e um produto final diferente daquele pretendido pelo cliente.

A etapa de coleta e a análise dos requisitos, sejam eles funcionais ou não funcionais, são tarefas complexas e importantes para o desenvolvimento do sistema. Caso ocorra falha nesta etapa, resultará em uma gama de erros decorrentes da falha inicial, pois ela se torna a base para todas as outras fases que serão desenvolvidas no sistema.

A interação do cliente com os analistas é de fundamental importância para a realização de um processo de requisitos bem definido, para que sejam identificados requisitos necessários ao sistema. Em seguida ocorre o processo de documentação dos requisitos levantados, à validação dos mesmos junto ao cliente para comprovar sua necessidade e viabilidade dentro do sistema. PFLEEGER (2004), afirma que assim se garante que a documentação é realmente o que o cliente deseja que se realize no sistema.

Dorfman e Thayer (1990), definem requisitos como “uma capacidade de software que deve ser atendida ou possuída por um sistema ou componentes do sistema para satisfazer um contrato, padrão, especificação ou outros documentos formalmente impostos.”

Os requisitos podem ser classificados em 2 tipos: requisitos funcionais e não funcionais.

4.1.1 Requisitos Funcionais

São descrições de funções do sistema, definem o seu comportamento do ao processar entradas específicas, ou seja, a interação do sistema com seu ambiente funcional. Eles têm o papel especificar quais tipos de entrada o sistema não deverá processar.

Devem-se especificar os requisitos funcionais de forma completa e consistente, para que seu objetivo possa ser o mais específico possível e não possibilite uma interpretação ambígua ou contraditória para o usuário.

Para descrever os requisitos, deve-se utilizar uma abordagem independente de uma solução específica de implementação para o problema do cliente. Pois a mesma deve conter uma camada de abstração referente à linguagem de programação, em qual computador será executado e quais estruturas de dados serão usadas.

4.1.2 Requisitos não funcionais

Diferentes dos requisitos funcionais, eles são as restrições oferecidas pelo sistema, que aborda os pontos que não possuem interação direta com os usuários. Porém são de extrema importância para a aceitação do software.

Eles estão relacionados com o tempo de resposta do sistema, a confiabilidade das informações solicitadas, segurança para seus usuários e também podem estar relacionados ao espaço em disco. O que se torna fundamental para que a aplicação funcione de maneira satisfatória. Também poderão ocorrer restrições de periféricos de entrada e saída para proverem escalabilidade, portabilidade e segurança.

Os requisitos apontados como não funcionais estão diretamente referenciando, na maioria das vezes, o sistema como um todo, deixando de lado as características individuais do sistema para os requisitos funcionais. De acordo com Sommerville (2011), pode-se até dizer que muitos desses requisitos são mais importantes que os próprios requisitos funcionais, pois existem alguns requisitos que se não forem implantados, possivelmente tornarão o uso do sistema inviável.

4.2 Linguagem de Modelagem UML

O modelo de *UML* (do inglês, *Unified Modeling Language*) ou Linguagem de Modelagem Unificada que conhecemos hoje, surgiu da união de três métodos de modelagem, a partir dos trabalhos e pesquisas dos seguintes autores: método de Booch, de Grady Booch, método OMT (*Object Modeling Technique*) de Ivar

Jacobson e o método OOSE (*Object-oriented Software Engineering*) de James Rumbaugh.

Segundo Medeiro (2004), eles eram, por volta da década de 90, os três métodos de modelagem orientada a objetos mais utilizados por profissionais da área de desenvolvimento de software. Para a união dessas três metodologias de modelagem, contou-se com a grande participação da *Rational Software*, que apoiou e financiou a junção das três metodologias.

Define-se a UML como uma linguagem visual utilizada para modelar sistemas computacionais por meio do paradigma de Orientação a Objetos.

A UML tem como objetivo de ajudar a definir as características do software. Define como serão os seus requisitos, comportamento e estrutura lógica. Essas características são definidas por meio da UML antes do software começar a ser realmente desenvolvido, de acordo com Guedes (2004).

A estrutura da linguagem UML é composta por diferentes tipos de diagrama, cada diagrama representa uma determinada forma do sistema. Com a utilização de diversos diagramas permite-se que falhas sejam descobertas, diminuindo a possibilidade da ocorrência de erros futuros, o que resulta em uma maior produtividade e menor refatoração corretiva no software.

Para este trabalho, serão utilizados os diagramas de casos de uso, de classes e de atividades.

4.2.1 Diagrama de Caso de Uso

São visões gráficas das funcionalidades providas pelo sistema. Ele é constituído por atores, casos de uso e os relacionamentos entre eles. O principal objetivo deste diagrama é demonstrar e documentar o que o sistema faz do ponto de vista do usuário e também especificar as funções do sistema que são utilizadas por um ator.

Utiliza-se a seguinte notação, segundo Sommerville (2011):

- O ator é representado por um boneco, com o seu respectivo nome embaixo. O ator pode ser outro sistema computacional ou um cliente do sistema (Figura 3 abaixo).

- Para o caso de uso utiliza-se uma elipse como representação, dentro está o nome do caso de uso. Assim o caso de uso é a demonstração de uma funcionalidade do sistema (Figura 4 abaixo).
- Nos relacionamentos utilizam-se setas com características associadas ao tipo de relacionamento. Podem-se ter os seguintes tipos de relacionamentos:
 - Entre um caso de uso e um ator, é definido como associação, um caso de uso está associado a um ator por uma linha contínua, ilustrada na Figura 4 abaixo.

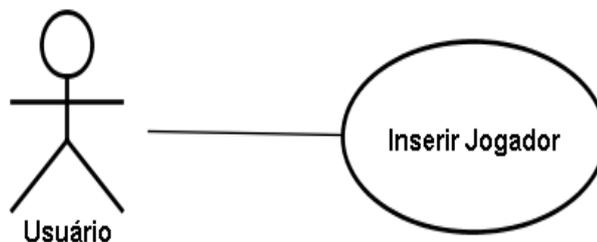


Figura 4 - Relacionamento de associação entre o ator e o caso de uso.

Fonte: SOMMERVILLE (2011)

- Entre dois atores ou dois casos de uso pode ser definida uma generalização, que ocorre quando um caso de uso / ator é herdeiro do outro caso de uso / ator. Esta representação é demonstrada na figura 5 abaixo.

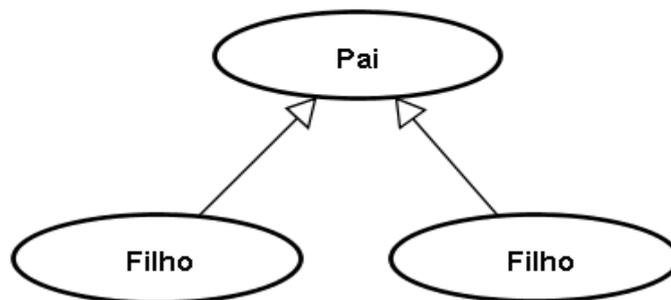


Figura 5: Relacionamento de generalização entre dois casos de uso.

Fonte: SOMMERVILLE (2011)

- Entre dois casos de uso, o subtipo *include*, é o caso de uso que está incluso de maneira obrigatória em outro caso de uso, ou seja, ao se executar um caso de uso no qual o outro está

incluso, ele também será executado. O subtipo *include* é representado pela Figura 6 abaixo.

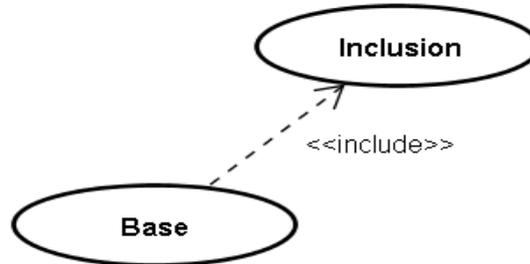


Figura 6: Relacionamento de inclusão entre dois casos de uso.

Fonte: SOMMERVILLE (2011)

- Ou subtipo *extend*, é o caso de uso de extensão de outro caso de uso de maneira opcional, ou seja, ao ser executado o caso de uso no qual o outro caso de uso é extensão, pode-se ou não executar este. O subtipo *extend* é representado pela Figura 7 abaixo.

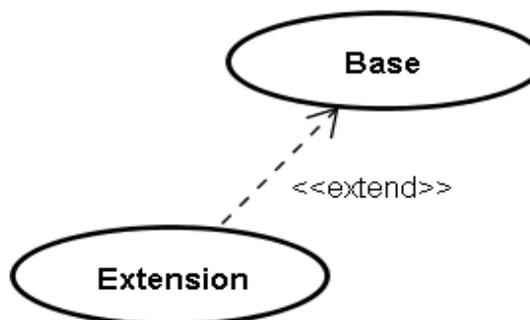


Figura 7: Relacionamento de extensão entre dois casos de uso.

Fonte: SOMMERVILLE (2011)

4.2.2 Diagrama de Classes

O diagrama de classes representa a estrutura do sistema, é constituído por classes, atributos e relacionamentos com outras classes. É um diagrama muito importante, pois ele fornece uma visão ampla dos relacionamentos do sistema.

De acordo com Guedes (2011), define-se classe como um elemento abstrato, que representa um objeto com suas características, ou seja, atributos e métodos. Seu diagrama é representado por um retângulo que se divide em três partes (Figura

8 abaixo), a primeira parte é constituída pelo nome da classe, a segunda é composta por seus atributos e a terceira encontram-se seus métodos.

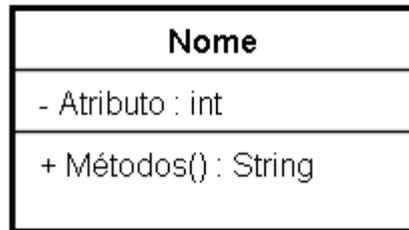


Figura 8: Nomenclatura de uma classe no Diagrama de Classes.

Fonte: GUEDES (2011).

Classifica-se a característica de uma classe quanto a sua visibilidade, ainda segundo Guedes (2011):

- **Pública:** está visível para todas as classes. Esta característica de classe é utilizada para representar um tipo público, usa-se o símbolo “+” juntamente com o nome da característica.
- **Protegida:** está visível somente para a própria classe e para quem possui relação de herança. Utiliza-se para representar uma característica protegida o símbolo “#” juntamente com o nome da característica.
- **Privada:** está visível exclusivamente para própria classe. Utiliza-se para representar uma característica privada o símbolo “-” juntamente do nome da característica.

Os relacionamentos entre as classes estão classificados pelos seguintes tipos, conforme com Booch (2005) e Guedes (2011):

- **Dependência:** representa um relacionamento de utilização, reflete que uma classe faz uso das informações e serviços de outra classe, mas não necessariamente o inverso, conforme a Figura 9 abaixo.

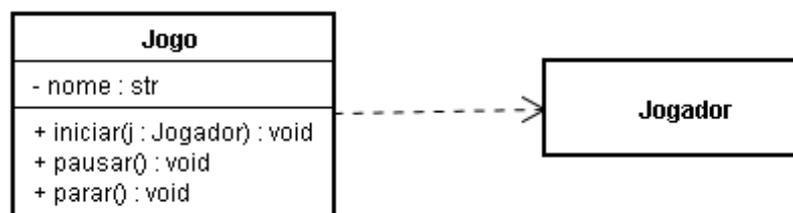


Figura 9: Relacionamento de dependência no diagrama de classes.

- Fonte: BOOCH (2005)

- **Associação:** representa as classes que possuem ligação, pois elas necessitam compartilhar informações entre si. A Figura 10 abaixo representa a notação de associação utilizada entre duas classes.

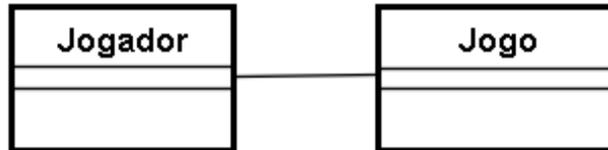


Figura 10: Relacionamento de associação no diagrama de classes.

Fonte: GUEDES (2011)

- **Generalização:** representa o relacionamento de herança entre classes, demonstra que uma subclasse herda os atributos e métodos de uma classe mais geral. A Figura 11 abaixo, representa a notação de generalização utilizada entre classes.

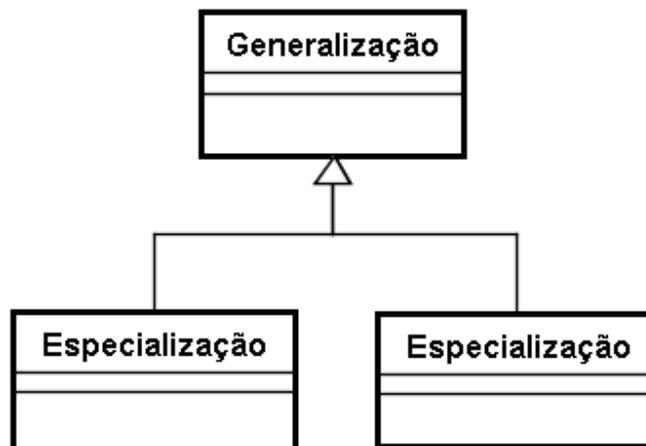


Figura 11: Relacionamento de generalização no diagrama de classes.

Fonte: GUEDES (2011)

- **Agregação:** representa que o objeto de uma classe precisa ser complementado com as informações de um objeto de outra classe. A Figura 12 abaixo, representa a notação de agregação utilizada entre classes.

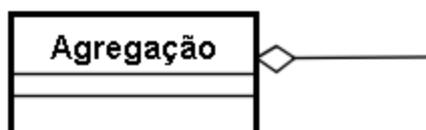


Figura 12: Relacionamento de agregação no diagrama de classes.

Fonte: GUEDES (2011)

- **Composição:** representa que o objeto de uma classe tem que ser composto com as informações de um objeto de outra classe. A Figura 13 abaixo representa a notação de composição utilizada entre classes.



Figura 13: Relacionamento de composição no diagrama de classes.

Fonte: GUEDES (2011).

Os relacionamentos são representados pela quantidade de objetos de uma classe no qual se pode relacionar com objetos de outras classes. Esses relacionamentos são chamados de multiplicidade do relacionamento. As multiplicidades são:

- 1, significa que o objeto de uma classe se relaciona obrigatoriamente com apenas um objeto de outra classe.
- 0..* ou *, significa que o objeto de uma classe se relaciona com zero ou muitos objetos de outras classes, constituindo assim um relacionamento não obrigatório.
- 0..1, significa que o objeto de uma classe pode se relacionar com zero ou um objeto da outra classe, constituindo assim um relacionamento não obrigatório.
- 1..*, significa que o objeto de uma classe se relaciona com um ou mais objetos de outras classes, constituindo assim um relacionamento obrigatório, ou seja, ele deve se relacionar com pelo menos um objeto de outra classe.

4.2.3 Diagrama de Atividades

Define-se o diagrama de atividades como uma visão gráfica do fluxo de atividades de um processo. Ele é constituído por ações, nós e fluxos, as atividades são divididas em regiões que são chamados de *swimlanes* (são raias que dividem o diagrama de atividades, identificando os agentes responsáveis por determinadas atividades) associadas a um objeto.

Já a ação é uma descrição de um comportamento que é responsável pela conversão de um conjunto de entradas em um conjunto de saídas, sejam estes conjuntos vazios ou não.

A notação, segundo Booch (2005), é representada por um retângulo de bordas arredondadas conforme a Figura 14 a seguir.

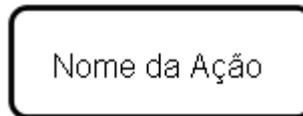


Figura 14: Representação de uma ação no diagrama de atividades.

Fonte: BOOCH (2005).

Ocorrem casos em que as ações recebem um evento. Para estes casos, a notação empregada é representada na Figura 15 abaixo.



Figura 15: Representação de uma ação que recebeu um evento no diagrama de atividades.

Fonte: BOOCH (2005).

Se o evento for temporal, utiliza-se uma ampulheta, de acordo com a Figura 16 abaixo.

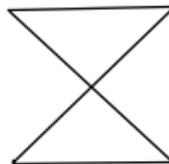


Figura 16: Representação de uma ação que recebe um evento temporal no diagrama de atividades.

Fonte: BOOCH (2005).

Os nós têm a responsabilidade de coordenar o fluxo da atividade e são classificados pelos seguintes tipos a seguir.

O nó inicial indica o início de um fluxo, todos os nós iniciais de uma atividade serão invocados no seu início. Sua presença é facultativa. E sua representação é um círculo preto de acordo com a Figura 17 abaixo.



Figura 17: Representação de um nó inicial no diagrama de atividades.

Fonte: BOOCH (2005).

O nó final de atividade que determina o fim da atividade, ao terminar todos os caminhos paralelos que são seguidos. É representado por um círculo preto dentro de um círculo vazado, de acordo com a Figura 18 abaixo.



Figura 18: Representação de um nó final de atividade no diagrama de atividades.

Fonte: BOOCH (2005)

O nó final de fluxo indica que o caminho seguido terminou. Porém ele não representa o fim da atividade, se por acaso outros caminhos sejam seguidos. Sua representação é mostrada na Figura 19 abaixo.



Figura 19: Representação de um nó final de fluxo no diagrama de atividades.

Fonte: BOOCH (2005)

O nó de *fork* divide um fluxo em vários fluxos concorrentes. Sua representação gráfica é mostrada na Figura 20 abaixo.



Figura 20: Representação do nó *fork* no diagrama de atividades.

Fonte: BOOCH (2005).

O nó de *join* unifica e sincroniza os fluxos concorrentes. A sua notação é semelhante a do nó *fork* (figura 20 acima).

O nó de decisão permite escolher um entre vários caminhos possíveis. Devem-se conter expressões lógicas para os caminhos, são chamadas de “guardas”, que proporcionam a decisão de qual caminho tomar. Os “guardas” devem estar dentro de colchetes. O losango representa o nó de decisão, de acordo com a Figura 21 abaixo.



Figura 21: Representação de um nó de decisão no diagrama de atividades.

Fonte: BOOCH (2005).

O nó de *merge* serve para unir os caminhos que foram separados pelo nó de decisão. A sua representação é a mesma do nó de decisão.

O nó de objeto representa um objeto e é montado em fluxos de objeto entre duas ações. A sua representação é uma caixa retangular, de acordo com a Figura 22 abaixo.

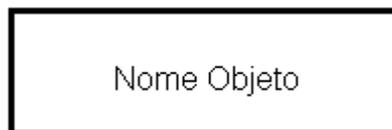


Figura 22: Representação de um nó de objeto no diagrama de atividades.

Fonte: BOOCH (2005)

Já os objetos, que são sinais, possuem uma notação especial, de acordo com a Figura 23 abaixo.

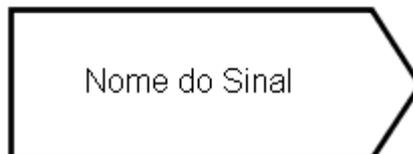


Figura 23: Representação de um nó de objeto que é sinal no diagrama de atividades.

Fonte: BOOCH (2005)

4.3 Modelagem Relacional

O Modelo Relacional dos dados é representado como um conjunto de relações. Considerando que uma relação é, de certo modo, similar a uma tabela de valores e aplicando a terminologia do Modelo Relacional, diz-se que as linhas denominam-se tuplas; as colunas, atributos; e a tabela em si, relação (ELMASRI; NAVATHE, 2011).

O conjunto de *tuplas* (linhas da tabela), também representam uma entidade ou relacionamento do mundo real, juntamente com seus respectivos valores para cada atributo.

O conjunto de valores que cada atributo pode assumir chama-se domínio e uma relação é definida matematicamente como um subconjunto do produto cartesiano de uma lista de domínios, segundo Silberschatz et al. (1999).

Já na estrutura interna de cada tabela devemos definir todos os atributos, que devam ser atômicos, de acordo com Elmasri e Navathe (2011). Citam que atributos não divisíveis são chamados atributos atômicos, e que não possam ter mais de uma tupla com os mesmos valores, pode-se ter chaves candidatas.

Para os atributos que são chaves candidatas é obrigatório que não sejam nulos, assim proporciona que sejam utilizados para identificar individualmente uma tupla, que possui a nomenclatura de chave primária.

O relacionamento entre tabelas ocorre quando uma tabela necessita de informações de outra tabela. Então se deve acrescentar o atributo, que irá conter as informações necessárias, à tabela que necessita da informação da outra, é chamado esse atributo de chave estrangeira.

Para se obter o modelo relacional se faz necessário uma conversão do diagrama de classes, a conversão é demonstrada na próxima seção.

4.3.1 Conversão de diagrama de classes em modelo relacional

De acordo com Elmasri e Navathe(2011), para converter um diagrama de Classes em modelo relacional existem regras necessárias e alguns passos a serem seguidos:

- Cada classe é mapeada para uma tabela.
- Cada atributo da classe é mapeado para coluna da tabela correspondente a classe.
- Cada método da classe é mapeado para uma *stored procedure* ou para funções externas ao banco de dados.
- Para executar a conversão usa-se uma chave primária formada por um atributo inteiro sequencial (auto incremento), usualmente chamada de ID, para cada classe mapeada.

- Já para os relacionamentos do tipo associação, agregação e composição, não há diferença no seu mapeamento. O mapeamento será feito de acordo com sua cardinalidade:
 - Para (0,1) – (0,1), cada tabela terá sua própria chave primária e uma das tabelas terá uma chave estrangeira para a outra.
 - Para (0,1) – (1,1) e (0,*) – (1,1), pode-se ter:
 - Cada tabela tendo sua própria chave primária e a tabela, o relacionamento é obrigatório, e possui uma chave estrangeira para a outra tabela.
 - Ou cada tabela tem sua própria chave primária e na tabela, o relacionamento é obrigatório, a chave primária é uma chave estrangeira para a outra também.
 - Ou cada tabela tem sua própria chave primária, visto que a chave primária da tabela, possui relacionamento obrigatório, é composta por uma chave estrangeira para a outra tabela.
 - Para (0,1) – (0,*), cada tabela possui sua própria chave primária e a tabela, o relacionamento pode ser (0,*), possui uma chave estrangeira para a outra tabela.
 - Para (0,1) – (1,*), pode-se ter:
 - Cada tabela tendo sua própria chave primária e a tabela, pois o relacionamento pode ser (1,*), com uma chave estrangeira para a outra tabela.
 - Ou cada tabela com sua própria chave primária, pois a chave primária da tabela, e o relacionamento pode ser (1,*), é composta por uma chave estrangeira para a outra tabela.
 - Para (0,*) – (0,*) e (0,*) – (1,*), cada tabela possui uma chave primária e é criada uma tabela para representar o relacionamento contendo chaves estrangeiras para as duas tabelas do relacionamento.
- Para relacionamentos do tipo herança, se dividem em três as opções para mapeamento:
 - É criada uma única tabela para toda a hierarquia da superclasse (classe pai), pois haverá um atributo que identificará a qual subclasse (classe filho) o objeto pertence e todos os atributos de todas as subclasses deverão estar nesta tabela.

- Através de fragmentação vertical, será criada uma tabela para cada subclasse da hierarquia com os atributos correspondentes. Neste caso, objeto é dividido em duas ou mais partes.
- Através de fragmentação horizontal, aplicadas apenas as subclasses concretas, ou seja, as que possuem instâncias serão representadas no modelo. Os atributos que existirem na superclasse serão repetidos nas tabelas que representam cada subclasse concreta.

4.4 Dicionário de Dados

O dicionário de dados é a descrição das entidades envolvidas no processo de desenvolvimento do sistema, conforme os autores Elmasri e Navathe (2011). Ele refina os requisitos funcionais do software.

Assim, o dicionário de dados é uma listagem com as definições de cada uma das informações lógicas a serem armazenadas no banco de dados, como citam Silberschatz et al. (1999).

Pode-se definir a notação de uma informação lógica como:

<termo a ser definido>=<definição do termo>

Onde a definição do termo pode assumir várias formas:

- Combinar outros termos, a notação é a seguinte:

<termo>+<termo>+...

- Ao se repetir termos, se tem um número limite mínimo de repetições e um limite máximo de repetições. Possuem duas notações possíveis, uma abrange os casos em que não se possui limite mínimo e nem limite máximo, e outra quando se tem limite. As situações estão representadas a seguir:

{<termo>}

limite mínimo : limite máximo{<termo>}

- Termos opcionais, tendo como notação:

(<termo>)

- Uma escolha entre termos, sendo sua notação:

[<termo1>|<termo2>|...]

- Os valores possíveis, onde sua notação é:

“valor”

- Comentários, tendo como notação:

comentários

Utiliza-se comentários para itens simples. Já o significado do termo caso ele seja óbvio, utiliza-se:

dado elementar

Utiliza-se do símbolo “@” para identificar os termos componentes de chaves de outros termos.

5 MODELAGEM DA API GAMIFICADA

Neste capítulo será apresentada a modelagem para funcionamento da solução de API Gamificada, conforme proposta neste trabalho.

5.1 Contextualização e Coleta de Dados

Para a coleta dos requisitos foram observados vários elementos de Games compostos em algumas aplicações Gamificados no Web como: Livemocha, Foursquare e Duolingo.

O Livemocha é a maior comunidade mundial de aprendizagem de idiomas online. Ele combina métodos tradicionais de aprendizagem com a prática online e interação com falantes nativos ao redor do mundo. Possui mais de 15 milhões de membros em mais de 195 países. A sua Gamificação é baseada no sistema de pontos. Segundo Quadros (2011), é o que prepondera entre os usuários, pois à medida que avançam em suas lições, eles vão adquirindo pontos que elevarão o grau de representatividade dentro da comunidade de alunos e recebem de prêmios, como a liberação de atividades complementares da língua que está aprendendo. Se o aluno atuar significativamente, auxiliando alunos de outros países a aprender sua língua materna, ele conquista maior visibilidade como professor. Além disso, ele é condecorado com medalhas de reconhecimento, conforme a pontuação advinda dessa ajuda oferecida.

O Foursquare é uma rede social onde as pessoas fazem *check-in*, compartilham e dão dicas sobre lugares. O uso da ferramenta é incentivado com pontuações, ranking e *badges*, premiando os usuários quando se alcança um objetivo. O serviço é um dos exemplos mais conhecidos de Gamificação na Web e foi bem aceito pela população.

O Duolingo é uma aplicação desenvolvida para navegadores e dispositivos móveis que tem o objetivo de ensinar línguas estrangeiras. Para motivar os seus usuários ela emprega a Gamificação, utilizando elementos de jogos como: objetivos, níveis e ranking, pois manter uma pessoa concentrada em um estudo a distância é um grande desafio e, com esse modelo, ele consegue manter o aluno imerso por mais tempo que o usual, atingindo através do jogo o seu objetivo de transmitir o conteúdo.

Abaixo, na Figura 24, é apresentada uma interface do Duolingo que mostra o momento que o usuário acaba de obter uma pontuação. Além disso, ela mostra também as fases ou módulos que o usuário deverá percorrer durante o aprendizado.



Figura 24: interface de pontuação e fases do Duolingo

Fonte: www.duolingo.com

Já na interface apresentada na Figura 25, pode ser visualizada o desempenho dos usuários, emblemas obtidos e o ranking.



Figura 25: interface com desempenho, emblemas e ranking do usuário no Duolingo.

Fonte: www.duolingo.com

Por fim, a interface na Figura 26, apresenta de forma bem estruturada o fluxo de lições para o usuário evoluir de fase ou módulo em um determinado conteúdo dentro do ambiente Duolingo.

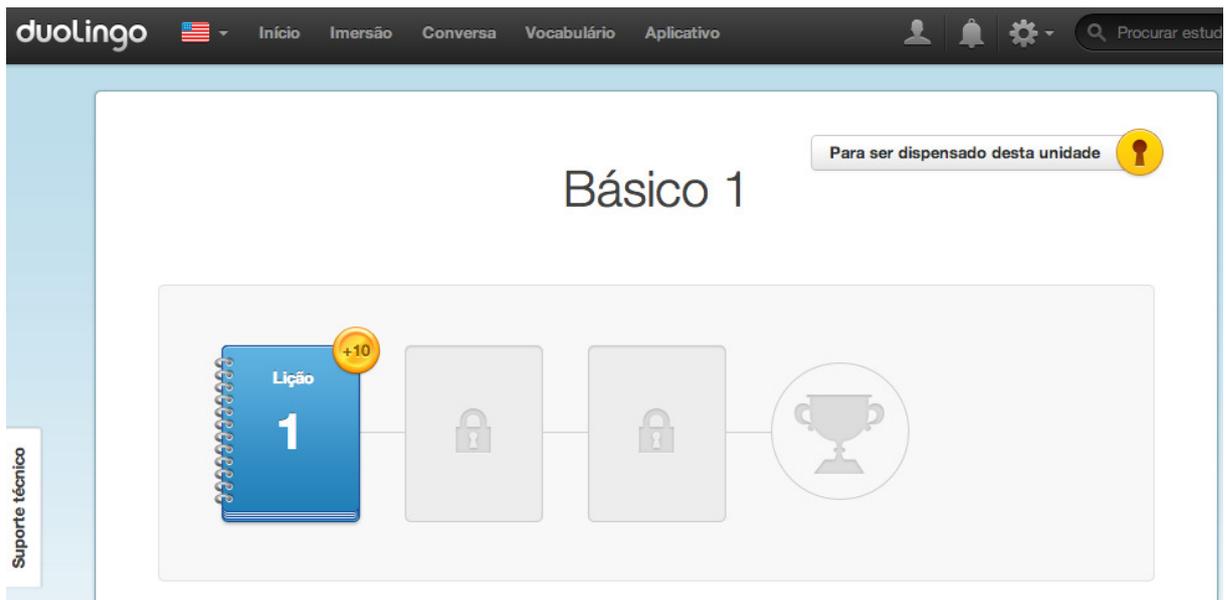


Figura 26: interface de lições do Duolingo

Fonte: www.duolingo.com

Diante dos ambientes gamificados observados e da literatura especializada, apresentada no Capítulo 02, definiram-se os requisitos da solução da API Gamificada. Ela foi modelada com o objetivo de disponibilizar serviços com conceitos de Gamificação para integração de aplicações ou ambientes na Web de terceiros ou parceiros. A solução proposta neste trabalho é composta por 2 módulos: o módulo web Administrador e a API Gamificada.

Nos próximos tópicos serão apresentados o Diagrama de classes, a modelagem do módulo web Administrador, a modelagem da API Gamificada e o Diagrama de Entidade Relacionamento da solução proposta.

5.2 Diagrama de Classe dos Módulos

A seguir é apresentado o diagrama de classes conceitual modelado para a solução da API Gamificada. Neste diagrama são utilizados alguns dos tipos de relacionamento apresentados anteriormente no Capítulo 4, o relacionamento de associação, dependência e composição de acordo com a Figura 27.

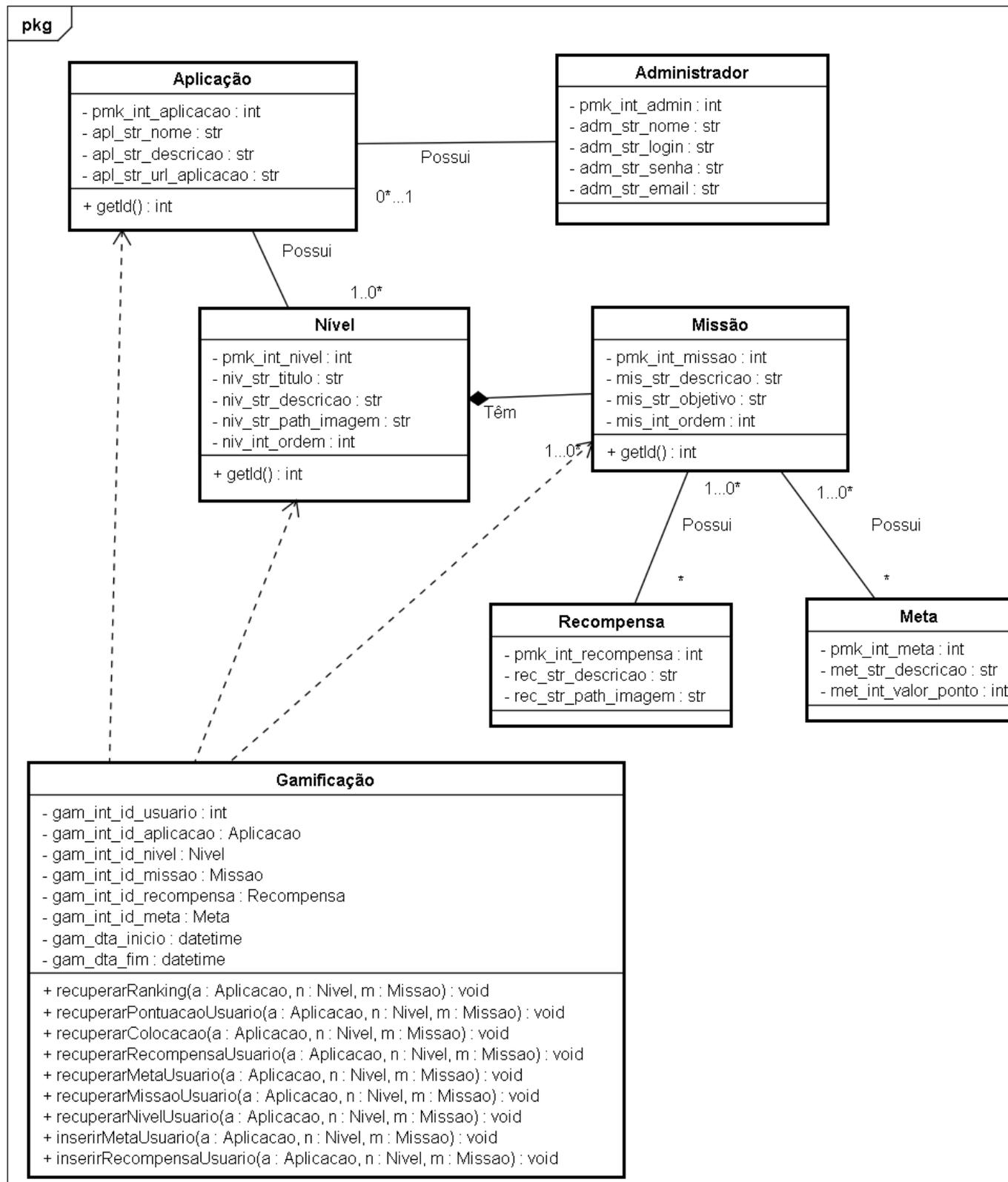


Figura 27 - Diagrama de Classes
Fonte: Elaboração própria do autor

Conforme a Figura 27, abaixo estão representadas as descrições do diagrama de classes criado, no qual apresenta-se a descrição das classes de forma textual.

- **Administrador:** é a classe que representa os atributos do usuário que irá gerenciar os dados do usuário Administrador e os métodos que ele poderá efetuar. Os métodos da classe permitem a consulta e configurações do perfil do usuário, ele será o responsável pela configuração e manutenção dos dados básicos que servirão para o correto funcionamento da API Gamificada integrada a sua aplicação. Esta classe possui uma relação de associação com a classe Aplicação, estruturando Níveis, Missões, Metas e Recompensas do sistema;
- **Aplicação:** esta classe armazena as informações das aplicações de terceiros ou parceiros que serão gamificadas, descrevendo as características da aplicação do cliente. Ela será a classe responsável pela estrutura geral da aplicação. Um usuário administrador podem ter várias aplicações cadastradas e, conseqüentemente, gamificadas. Possui uma relação de associação com a classe Nível, onde cada aplicação pode ser composta por vários níveis;
- **Nível:** classe responsável por definir a hierarquia dos usuários da aplicação do cliente dentro da estrutura gamificada. Através desta classe é que são definidos os níveis que um usuário da aplicação do cliente já conquistou e o que pode ser alcançado. Possui uma relação de composição com a classe Missão. Ela permite a exibição dos usuários de forma sistemática de acordo com sua evolução dentro da estrutura Gamificada. Cada aplicação podem ter vários níveis, como por exemplo:
 - a. Nível aprendiz;
 - b. Nível auxiliar;
 - c. Nível especialista;
- **Missão:** define a quantidade e sequências das missões (atividades com objetivos definidos) que o usuário da aplicação do cliente deverá cumprir na aplicação gamificada. Indica também qual o objetivo será proposto para o usuário da aplicação do cliente, sendo dividido em duas vertentes, uma está vinculada a recompensas que se deve ganhar e a outra é referida às metas que devem ser alcançadas. Cada nível podem ter várias missões, como por exemplo:

- a. Missão 01 – acessar o sistema 90 vezes durante 1 mês;
 - b. Missão 02 – interagir via mensagem com mais de 5 usuários em um período de 24 horas;
 - c. Missão 03 – contribuir no fórum com um artigo sobre uma determinada área de trabalho da empresa.
- **Recompensa:** esta classe é composta pelas recompensas que a aplicação do cliente gamificada possuirá para fornecer aos seus usuários. Possui uma relação de associação com a classe missão. Ela representa as premiações que o usuário receberá ao conseguir realizar alguma missão dentro da aplicação do cliente. Cada missão pode possuir várias recompensas, como por exemplo:
 - a. Recompensa 01 – Possibilidade de mudar as cores de fundo do sistema;
 - b. Recompensa 02 – Possibilidade de compartilhar as suas conquistas no Facebook;
 - c. Recompensa 03 – Participação como moderador em um fórum de discussão no sistema;
- **Meta:** representam os objetivos mais específicos, que compõem um objetivo geral (missão), e as pontuações disponíveis no sistema. Ela é alcançada quando o usuário da aplicação do cliente consegue cumprir as tarefas definidas para uma missão, através desses pontos e associados com as recompensas, o usuário consegue alcançar outros níveis dentro da solução gamificada. Ela possui uma relação de associação com a classe missão. Cada missão pode possuir várias metas, como por exemplo:
 - a. Meta 01 - acessar o sistema 3 vezes diariamente durante 10 dias em um mês;
 - b. Meta 02 – acessar o sistema 3 vezes diariamente durante 20 dias em um mês;
 - c. Meta 03 – acessar o sistema 3 vezes diariamente durante 30 dias em um mês;
- **Gamificação:** classe responsável por gerenciar o processo de gamificação na aplicação do cliente. Ela permite o armazenamento ou registro das metas, pontuações e recompensas alcançadas pelos usuários na aplicação do cliente. Possibilita também a recuperação das principais informações dos

usuários via API Gamificada. Ela possui uma relação de dependência com as classes de Aplicação, Nível e Missão. Com esta classe é possível registrar os dados como as recompensas e metas alcançadas pelos usuários, além de recuperar informações sobre:

- a. Ranking dos usuários de forma geral ou por aplicação, missão, nível ou período;
- b. Pontuação dos usuários de forma geral ou por aplicação, missão, nível ou período;
- c. Colocação de um usuário de forma geral ou por aplicação, missão, nível ou período;
- d. Missões, recompensas e metas alcançadas pelos usuários de forma geral ou por aplicação, missão, nível ou período;

5.3 Módulo Web do Administrador

É o módulo web em que o usuário desenvolvedor ou mantenedor de uma aplicação cliente configurará os seus dados e os itens de Gamificação como Aplicação, Níveis, Missões, Recompensas e Metas. Neste módulo, o usuário administrador deverá se cadastrar, criando uma conta de usuário e, a partir disso, cadastrar aplicações e as informações necessárias para posterior uso da API Gamificada.

5.3.1 Diagrama de Casos de Uso

A Figura 28 abaixo apresenta o diagrama principal de caso de uso elaborado na modelagem do Módulo Web do Administrador. Este diagrama apresenta as funcionalidades deste módulo de maneira geral.

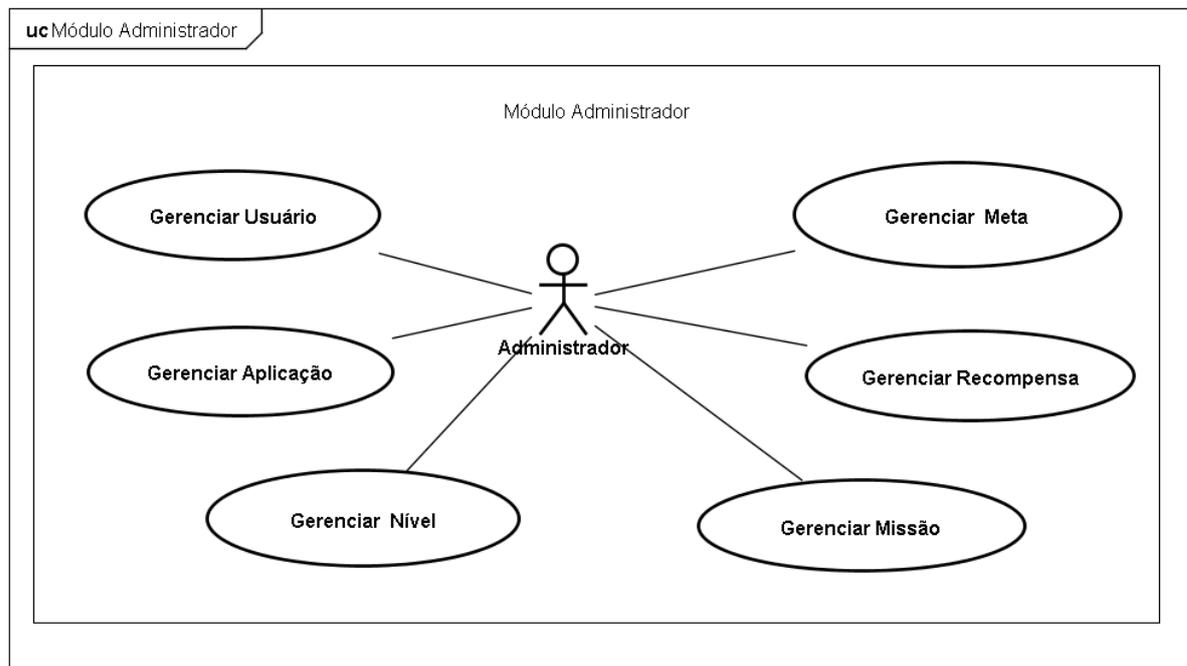


Figura 28: Caso de uso principal módulo Web do Administrador

Fonte: Elaboração própria do autor

Todos os casos de usos, citados na Figura 28 acima, são compostos por casos de uso mais específicos, conforme pode ser visualizado abaixo. Por este motivo necessitam ser mais detalhados. Em seguida são apresentados os fluxos de eventos para cada caso de uso.

Na Figura 29 abaixo são representados os casos de uso que compõem o caso de uso "Gerenciar Usuário";

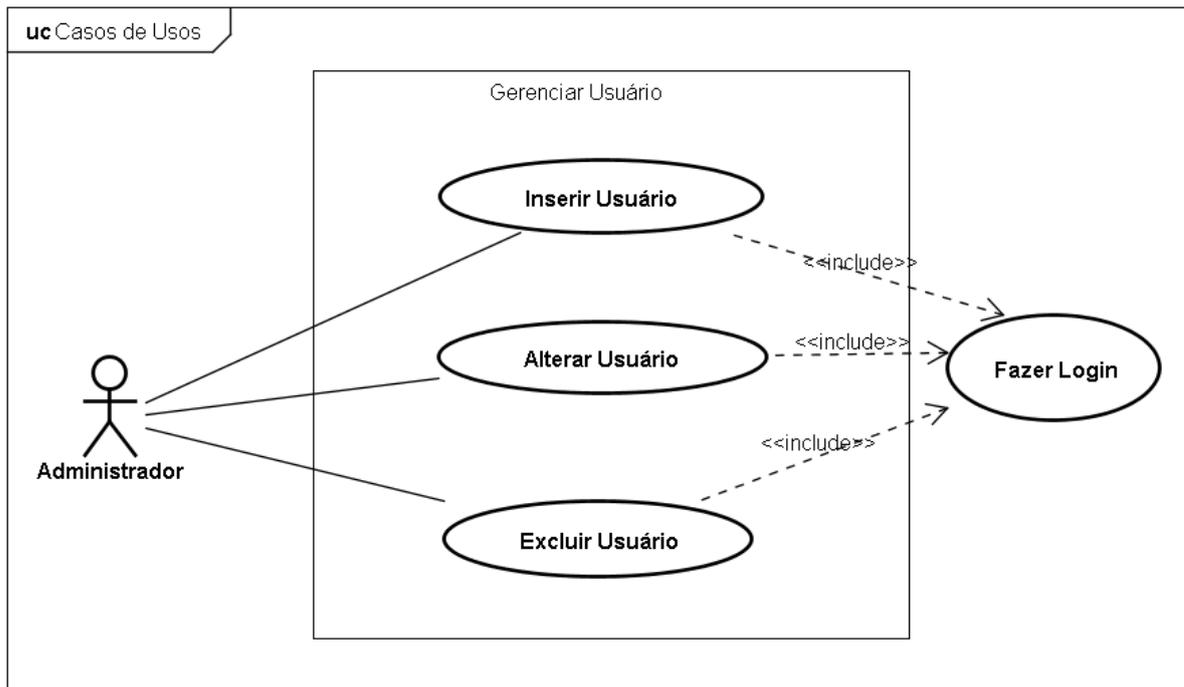


Figura 29: Caso de uso Gerenciar Usuário

Fonte: Elaboração própria do autor

Na Figura 30 abaixo são representados os casos de uso que compõem o caso de uso "Gerenciar Aplicação".

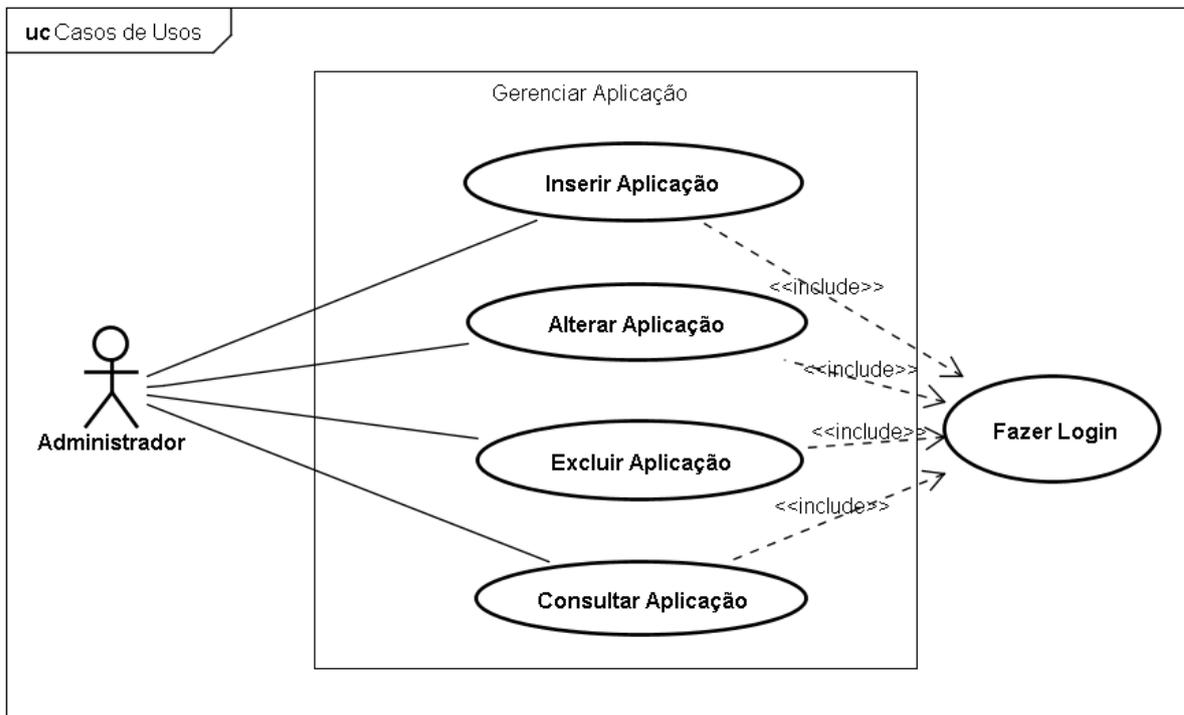


Figura 30: Caso de uso Gerenciar Aplicação

Fonte: Elaboração própria do autor

Na Figura 31 abaixo são representados os casos de uso que compõem o caso de uso "Gerenciar Nível".

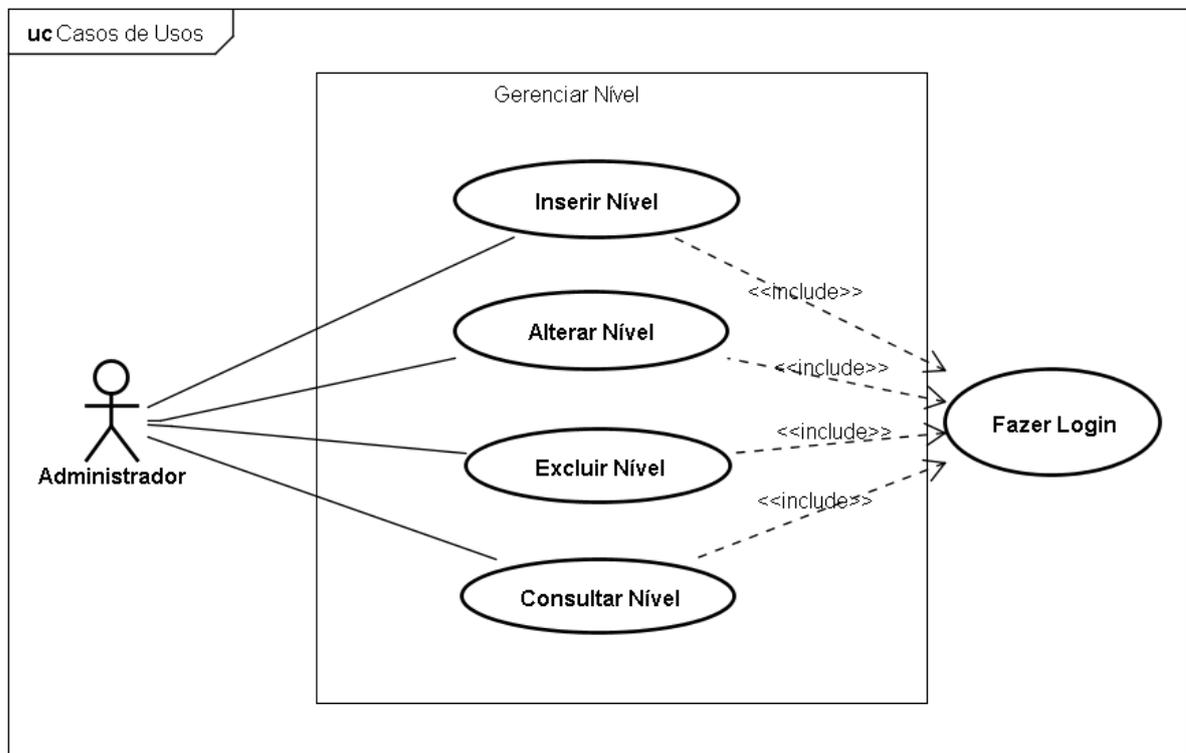


Figura 31: Caso de uso Gerenciar Nível

Fonte: Elaboração própria do autor

Na Figura 32 abaixo são representados os casos de uso que compõem o caso de uso "Gerenciar Missão".

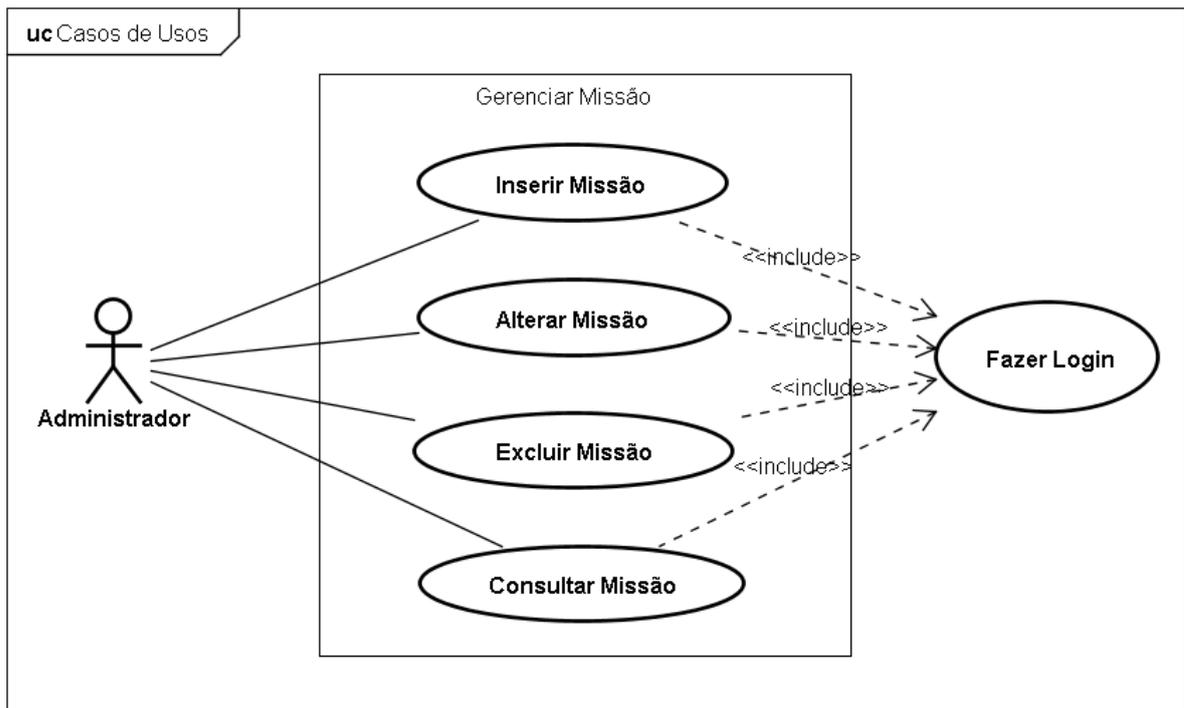


Figura 32: Caso de uso Gerenciar Missão

Fonte: Elaboração própria do autor

Na Figura 33 abaixo são representados os casos de uso que compõem o caso de uso "Gerenciar Recompensa".

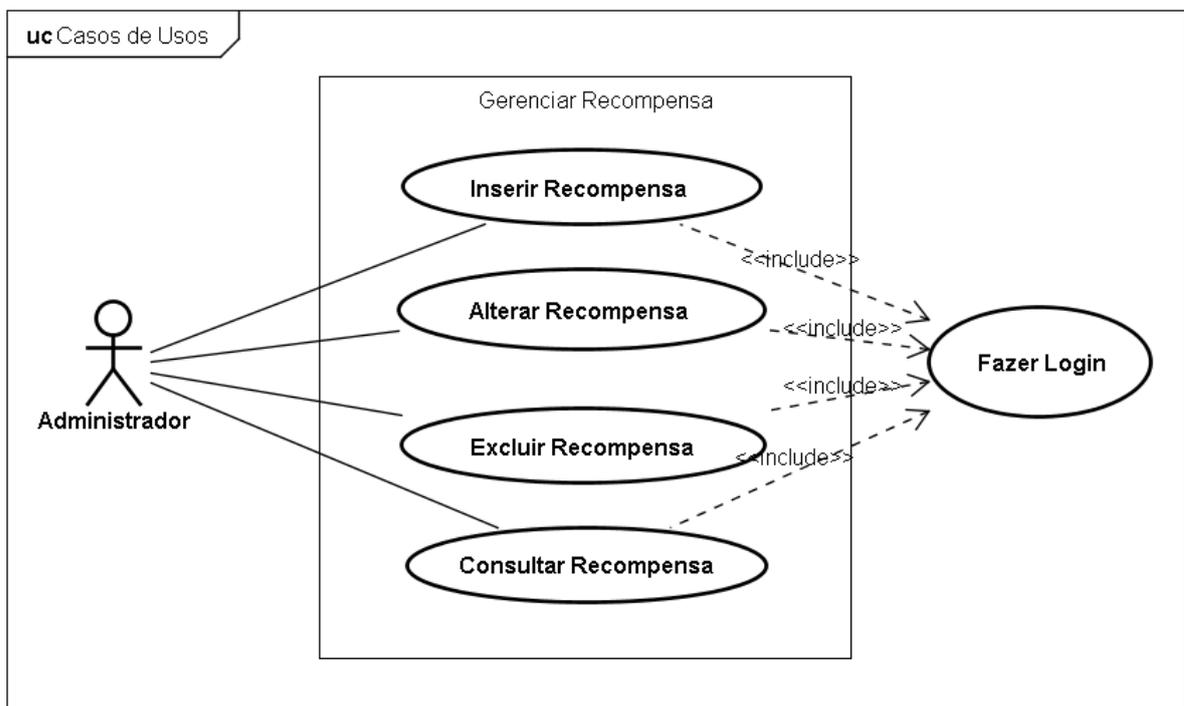


Figura 33: Caso de uso Gerenciar Recompensa

Fonte: Elaboração própria do autor

Na Figura 34 abaixo são representados os casos de uso que compõem o caso de uso "Gerenciar Meta".

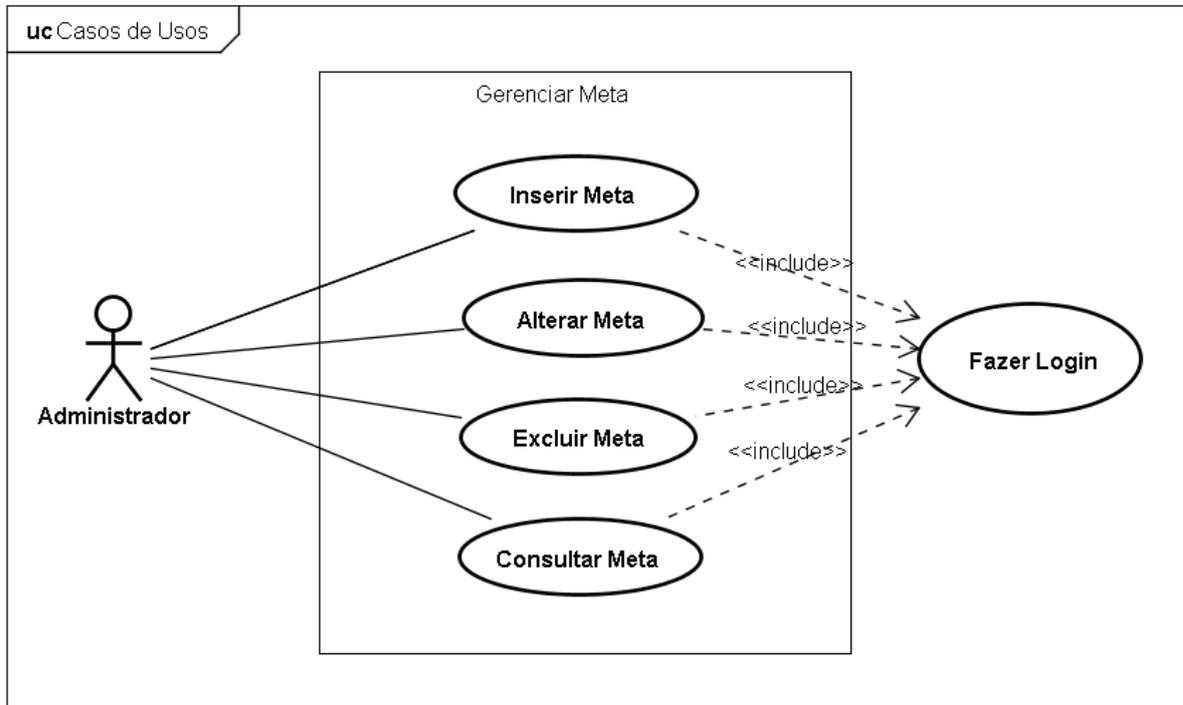


Figura 34: Caso de uso Gerenciar Meta

Fonte: Elaboração própria do autor

5.3.2 Requisitos Funcionais

Abaixo serão apresentados os fluxos de eventos para cada caso de uso citado no sub-tópico anterior.

5.3.2.1 Caso de uso Fazer Login

Devido o caso de uso "Fazer Login" ser o mesmo para todos os casos de uso citados, ele será apresentado apenas uma vez.

- **Descrição:** este caso de uso aborda o procedimento adotado para ter acesso aos recursos do sistema.
- **Pré-condições:** o usuário já deve ter um cadastro e possuir login e senha previamente fornecidos pelo módulo Web do administrador, antes que este caso de uso se inicie.
- **Atores envolvidos:** administrador do sistema.

- **Fluxo principal:** este caso de uso se inicia quando o usuário deseja realizar login no módulo Web do administrador. Na sequência ele deve fornecer login e senha e selecionar a atividade desejada: fazer login ou sair. Se a atividade selecionada for fazer login, o sub-fluxo "fazer login" será executado. Se a atividade selecionada for Sair, o caso de uso termina.
- **Sub-fluxos:** "fazer login" - o sistema exibe uma janela para entrada dos seguintes dados: login e senha. O módulo Web do administrador valida os dados e o caso de uso é reiniciado.
- **Fluxos alternativos:** na tentativa de fazer login, caso o login ou senha sejam inválidos, o sistema exibirá uma mensagem informando tal situação. O usuário pode fornecer um novo login e senha ou terminar o caso de uso.

5.3.2.2 Caso de uso Inserir Usuário

- **Descrição:** este caso de uso aborda o procedimento adotado para o cadastro de usuários, no módulo Web do administrador, interessados em usar a API em suas aplicações Web.
- **Pré-condições:** nenhuma.
- **Atores envolvidos:** administrador do sistema.
- **Fluxo principal:** este caso de uso se inicia após o usuário acessar a interface inicial do módulo Web do administrador e clicar no botão cadastrar-se. Na sequência ele deve preencher os seguintes dados: nome do usuário, login, senha e e-mail. O sistema salva os dados e o caso de uso é reiniciado.
- **Sub-fluxos:** nenhum.
- **Fluxos alternativos:** na tentativa de cadastro de um usuário, caso o login ou e-mail já estiverem cadastrados, o sistema exibirá uma mensagem informando tal situação. O usuário pode fornecer um novo login e e-mail ou terminar o caso de uso.

5.3.2.3 Caso de uso Alterar Usuário

- **Descrição:** este caso de uso aborda o procedimento adotado para a alteração de dados do usuário cadastrado no sistema.
- **Pré-condições:** o caso de uso "fazer login" deve ser executado antes que este caso de uso se inicie.

- **Atores envolvidos:** administrador do sistema.
- **Fluxo principal:** este caso de uso se inicia após o administrador ter realizado login no sistema. Na sequência o usuário deve selecionar a atividade desejada: alterar dados do usuário ou sair. Se a atividade selecionada for alterar dados do usuário o sub-fluxo "alterar usuário" será executado. Se a atividade selecionada for sair, o caso de uso termina.
- **Sub-fluxos:** "alterar dados do usuário" - o sistema exibe uma janela para alteração dos dados: login, senha atual, nova senha. O sistema salva os dados e o caso de uso é reiniciado.
- **Fluxos alternativos:** quando houver uma tentativa de alteração com uma senha, login ou e-mail inválido. O usuário pode fornecer uma nova senha, login ou e-mail ou terminar o caso de uso.

5.3.2.4 Caso de uso Excluir Usuário

- **Descrição:** este caso de uso aborda o procedimento adotado para a exclusão de usuário no sistema.
- **Pré-condições:** o caso de uso "fazer login" deve ser executado antes que este caso de uso se inicie.
- **Atores envolvidos:** administrador do sistema.
- **Fluxo principal:** este caso de uso se inicia após o administrador ter realizado login no sistema. Na sequência ele deve selecionar a atividade desejada: excluir usuário ou sair. Se a atividade selecionada for excluir usuário, o sub-fluxo "excluir usuário" será executado. Se a atividade selecionada for sair, o caso de uso termina.
- **Sub-fluxos:** "excluir usuário" - o sistema apresenta uma janela para confirmação da exclusão do usuário. O sistema exibe o código e o nome do usuário para confirmação da exclusão. O sistema exclui o usuário e o caso de uso reinicia.
- **Fluxos alternativos:** nenhum.

5.3.2.5 Caso de uso Inserir Aplicação

- **Descrição:** este caso de uso aborda o procedimento adotado para o cadastro de aplicações de terceiros ou parceiros no módulo Web do Administrador.

- **Pré-condições:** o caso de uso "fazer login" deve ser executado antes que este caso de uso se inicie.
- **Atores envolvidos:** administrador do sistema.
- **Fluxo principal:** este caso de uso se inicia após o usuário ter realizado login no sistema. Na sequência ele deve selecionar a atividade desejada: inserir aplicação ou sair. Se a atividade selecionada for inserir aplicação, o sub-fluxo "inserir aplicação" será executado. Se a atividade selecionada for sair, o caso de uso termina.
- **Sub-fluxos:** "inserir aplicação" - o sistema exibe uma janela para entrada dos seguintes dados: nome da aplicação, descrição, *url* da aplicação. O sistema salva os dados e o caso de uso é reiniciado.
- **Fluxos alternativos:** na tentativa de cadastro de uma aplicação, caso a *url* já estiver cadastrada, o sistema exibirá uma mensagem informando tal situação. O usuário pode fornecer uma nova *url* ou terminar o caso de uso.

5.3.2.6 Caso de uso Alterar Aplicação

- **Descrição:** este caso de uso aborda o procedimento adotado para alteração de aplicações cadastradas no sistema.
- **Pré-condições:** o caso de uso "fazer login" deve ser executado antes que este caso de uso se inicie.
- **Atores envolvidos:** administrador do sistema.
- **Fluxo principal:** este caso de uso se inicia após o usuário ter realizado login no sistema. Na sequência, ele lista as aplicações cadastradas, o usuário seleciona a que deseja realizar o procedimento e depois seleciona a atividade desejada: alterar aplicação ou sair. Se a atividade selecionada for alterar aplicação, o sub-fluxo "alterar aplicação" será executada. Se a atividade selecionada for sair, o caso de uso termina.
- **Sub-fluxos:** "alterar aplicação" - o sistema exibe uma janela para alteração dos dados: nome, descrição e *url*. O sistema salva os dados e o caso de uso é reiniciado.
- **Fluxos alternativos:** houve uma tentativa de alteração de uma aplicação com *url* já cadastrada. O usuário pode fornecer uma nova *url* ou terminar o caso de uso.

5.3.2.7 Caso de uso Excluir Aplicação

- **Descrição:** este caso de uso aborda o procedimento adotado para a exclusão das aplicações cadastradas no sistema.
- **Pré-condições:** o caso de uso "fazer login" deve ser executado antes que este caso de uso se inicie.
- **Atores envolvidos:** administrador do sistema.
- **Fluxo principal:** este caso de uso se inicia após o usuário ter realizado login no sistema. Na sequência, ele lista as aplicações cadastradas, o usuário seleciona a que deseja realizar o procedimento e depois seleciona a atividade desejada: excluir aplicação ou sair. Se a atividade selecionada for excluir aplicação, o sub-fluxo "excluir aplicação" será executado. Se a atividade selecionada for sair, o caso de uso termina.
- **Sub-fluxos:** "excluir aplicação" - o sistema exibe o código e o nome da aplicação para confirmação da exclusão. O sistema exclui a aplicação e o caso de uso reinicia.
- **Fluxos alternativos:** nenhum.

5.3.2.8 Caso de uso Consultar Aplicação

- **Descrição:** este caso de uso aborda o procedimento adotado para a consulta das aplicações no sistema. O caso de uso se inicia quando o usuário deseja consultar uma aplicação no sistema e finaliza quando a consulta é concluída ou quando o usuário cancela o procedimento.
- **Pré-condições:** o caso de uso "fazer login" deve ser executado antes que este caso de uso se inicie.
- **Atores envolvidos:** administrador do sistema.
- **Fluxo principal:** este caso de uso se inicia após o usuário ter realizado login no sistema. Na sequência, ele lista as aplicações cadastradas, o usuário seleciona a que deseja realizar o procedimento e depois seleciona a atividade desejada: consultar aplicação ou sair. Se a atividade selecionada for consultar aplicação, o sub-fluxo "consultar aplicação" será executado. Se a atividade selecionada for sair, o caso de uso termina.

- **Sub-fluxos:** "consultar aplicação" - o sistema exibe o código e o nome da aplicação para confirmação da visualização. O sistema exibe as informações da aplicação.
- **Fluxos alternativos:** nenhum.

5.3.2.9 Caso de uso Inserir Nível

- **Descrição:** este caso de uso aborda o procedimento adotado para o cadastro de níveis de aplicação no sistema.
- **Pré-condições:** os casos de usos "fazer login" e "consultar aplicação" devem ser executados antes que este caso de uso se inicie.
- **Atores envolvidos:** administrador do sistema.
- **Fluxo principal:** este caso de uso se inicia após o usuário ter realizado login no sistema e consultado uma aplicação. Na sequência ele deve selecionar a atividade desejada: inserir nível ou sair. Se a atividade selecionada for inserir nível, o sub-fluxo "inserir nível" será executado. Se a atividade selecionada for sair, o caso de uso termina.
- **Sub-fluxos:** "inserir nível" - o sistema exibe uma janela para entrada dos seguintes dados: código da aplicação, título do nível, descrição, endereço da imagem e ordem. O sistema salva os dados e o caso de uso é reiniciado.
- **Fluxos alternativos:** na tentativa de cadastro de um nível, caso a ordem já estiver cadastrada para aquela aplicação, o sistema exibirá uma mensagem informando tal situação. O usuário pode fornecer uma nova ordem ou terminar o caso de uso.

5.3.2.10 Caso de uso Alterar Nível

- **Descrição:** este caso de uso aborda o procedimento adotado para a alteração de um nível de aplicação cadastrado no sistema.
- **Pré-condições:** os caso de usos "fazer login" e "consultar aplicação" devem ser executados antes que este caso de uso se inicie.
- **Atores envolvidos:** administrador do sistema.
- **Fluxo principal:** este caso de uso se inicia após o usuário ter realizado login e consultado uma aplicação no sistema. Na sequência, ele lista os níveis

cadastrados, o usuário seleciona o nível que deseja realizar o procedimento e depois seleciona a atividade desejada: alterar nível ou Sair. Se a atividade selecionada for alterar nível o sub-fluxo "alterar nível" será executado. Se a atividade selecionada for sair, o caso de uso termina.

- **Sub-fluxos:** "alterar nível" - o sistema exibe uma janela para alteração dos dados: código da aplicação, título do nível, descrição, endereço da imagem e ordem. O sistema salva os dados e o caso de uso é reiniciado.
- **Fluxos alternativos:** houve uma tentativa de alteração de um nível em ordem já cadastrada para uma determinada aplicação. O usuário pode fornecer um nova ordem do nível ou terminar o caso de uso.

5.3.2.11 Caso de uso Excluir Nível

- **Descrição:** este caso de uso aborda o procedimento adotado para a exclusão de nível em uma aplicação no sistema.
- **Pré-condições:** os casos de usos "fazer login" e "consultar aplicação" devem ser executados antes que este caso de uso se inicie.
- **Atores envolvidos:** administrador do sistema.
- **Fluxo principal:** este caso de uso se inicia após o usuário ter realizado login e consultada uma aplicação no sistema. Na sequência, ele lista os níveis cadastrados, o usuário seleciona o nível que deseja realizar o procedimento e depois seleciona a atividade desejada: excluir nível ou sair. Se a atividade selecionada for excluir nível, o sub-fluxo "excluir nível" será executado. Se a atividade selecionada for sair, o caso de uso termina.
- **Sub-fluxos:** "excluir nível" - o sistema exibe o código e o título do nível para confirmação da exclusão. O sistema exclui o nível e o caso de uso reinicia.
- **Fluxos alternativos:** nenhum.

5.3.2.12 Caso de uso Consultar Nível

- **Descrição:** este caso de uso aborda o procedimento adotado para a consulta dos níveis no sistema.
- **Pré-condições:** os casos de uso "fazer login" e "consultar aplicação" devem ser executados antes que este caso de uso se inicie.
- **Atores envolvidos:** administrador do sistema.

- **Fluxo principal:** este caso de uso se inicia após o usuário ter realizado login e consultada uma aplicação no sistema. Na sequência, ele lista os níveis cadastrados, o usuário seleciona o nível que deseja realizar o procedimento e depois seleciona a atividade desejada: consultar nível ou sair. Se a atividade selecionada for consultar nível, o sub-fluxo "consultar nível" será executado. Se a atividade selecionada for sair, o caso de uso termina.
- **Sub-fluxos:** "consultar nível" – o sistema exibe o código e o título do nível de uma aplicação para confirmação da visualização. O sistema exibe as informações do nível e o caso de uso reinicia.
- **Fluxos alternativos:** nenhum.

5.3.2.13 Caso de uso Inserir Missão

- **Descrição:** este caso de uso aborda o procedimento adotado para o cadastro de missões para um determinado nível em aplicações de terceiros ou parceiros no sistema.
- **Pré-condições:** os casos de uso "fazer login", "consultar aplicação" e "consultar nível" devem ser executados antes que este caso de uso se inicie.
- **Atores envolvidos:** administrador do sistema.
- **Fluxo principal:** este caso de uso se inicia após o usuário ter realizado login, consultada uma aplicação e um nível no sistema. Na sequência ele deve selecionar a atividade desejada: inserir missão ou sair. Se a atividade selecionada for inserir missão, o sub-fluxo "inserir missão" será executado. Se a atividade selecionada for sair, o caso de uso termina.
- **Sub-fluxos:** "inserir missão" - o sistema exibe uma janela para entrada dos seguintes dados: código do nível, descrição da missão, objetivo da missão e ordem da missão. O sistema salva os dados e o caso de uso é reiniciado.
- **Fluxos alternativos:** na tentativa de cadastro de uma missão, caso a ordem já estiver cadastrada, o sistema exibirá uma mensagem informando tal situação. O usuário pode fornecer uma nova ordem ou terminar o caso de uso.

5.3.2.14 Caso de uso Alterar Missão

- **Descrição:** este caso de uso aborda o procedimento adotado para a alteração de missão em um nível de uma determinada aplicação cadastrada no sistema.

- **Pré-condições:** os casos de uso "fazer login", "consultar aplicação", "consultar nível" devem ser executados antes que este caso de uso se inicie.
- **Atores envolvidos:** administrador do sistema.
- **Fluxo principal:** este caso de uso se inicia após o usuário ter realizado login no sistema. Na sequência, ele lista as missões cadastradas, o usuário seleciona a missão que deseja realizar o procedimento e depois seleciona a atividade desejada: alterar missão ou sair. Se a atividade selecionada for alterar missão o sub-fluxo "alterar missão" será executado. Se a atividade selecionada for sair, o caso de uso termina.
- **Sub-fluxos:** "alterar missão" - o sistema exibe uma janela para alteração dos dados: código do nível, descrição, objetivo e ordem. O sistema salva os dados e o caso de uso é reiniciado.
- **Fluxos alternativos:** houve uma tentativa de alteração de uma missão, caso a ordem já estiver cadastrada em um determinado nível, o sistema exibirá uma mensagem informando tal situação. O usuário pode fornecer uma nova ordem ou terminar o caso de uso.

5.3.2.15 Caso de uso Excluir Missão

- **Descrição:** este caso de uso aborda o procedimento adotado para a exclusão de missões em um nível de uma aplicação do sistema.
- **Pré-condições:** os casos de uso "fazer login", "consultar aplicação" e "consultar nível" devem ser executados antes que este caso de uso se inicie.
- **Atores envolvidos:** administrador do sistema.
- **Fluxo principal:** Este caso de uso se inicia após o usuário ter realizado login, consultada uma aplicação e um nível no sistema. Na sequência, ele lista as missões cadastradas, o usuário seleciona a missão que deseja realizar o procedimento e depois seleciona a atividade desejada: excluir missão ou sair. Se a atividade selecionada for excluir missão, o sub-fluxo "excluir missão" será executado. Se a atividade selecionada for sair, o caso de uso termina.
- **Sub-fluxos:** "excluir missão" - o sistema exibe o código e a descrição da missão para confirmação da exclusão. O sistema exclui a missão e o caso de uso reinicia.
- **Fluxos alternativos:** nenhum.

5.3.2.16 Caso de uso Consultar Missão

- **Descrição:** este caso de uso aborda o procedimento adotado para a consulta das missões no sistema.
- **Pré-condições:** os casos de uso "fazer login", "consultar aplicação" e "consultar nível" devem ser executados antes que este caso de uso se inicie.
- **Atores envolvidos:** administrador do sistema.
- **Fluxo principal:** este caso de uso se inicia após o usuário ter realizado login, consultada uma aplicação e um nível no sistema. Na sequência, ele lista as missões cadastradas, o usuário seleciona a missão que deseja realizar o procedimento e depois seleciona a atividade desejada: consultar missão ou Sair. Se a atividade selecionada for consultar missão, o sub-fluxo "consultar missão" será executado. Se a atividade selecionada for sair, o caso de uso termina.
- **Sub-fluxos:** "consultar missão" - o sistema exibe o código e a descrição da missão para confirmação da visualização. O sistema exibe as informações da missão e o caso de uso reinicia.
- **Fluxos alternativos:** nenhum.

5.3.2.17 Caso de uso Inserir Meta

- **Descrição:** este caso de uso aborda o procedimento adotado para o cadastro de metas em uma missão no sistema.
- **Pré-condições:** os casos de uso "fazer login", "consultar aplicação", "consultar nível" e "consultar missão" devem ser executados antes que este caso de uso se inicie.
- **Atores envolvidos:** administrador do sistema.
- **Fluxo principal:** este caso de uso se inicia após o usuário ter realizado login, consultado uma aplicação, um nível e uma missão no sistema. Na sequência, ele deve selecionar a atividade desejada: inserir meta ou sair. Se a atividade selecionada for inserir meta, o sub-fluxo "inserir meta" será executado. Se a atividade selecionada for sair, o caso de uso termina.
- **Sub-fluxos:** "inserir meta" - o sistema exibe uma janela para entrada dos seguintes dados: descrição da meta e valor da meta. O sistema salva os dados e o caso de uso é reiniciado.

- **Fluxos alternativos:** o usuário pode terminar o caso de uso sem cadastrar nenhuma meta.

5.3.2.18 Caso de uso Alterar Meta

- **Descrição:** este caso de uso aborda o procedimento adotado para a alteração de metas cadastradas em uma missão no sistema.
- **Pré-condições:** os casos de uso "fazer login", "consultar aplicação", "consultar nível" e "consultar missão" devem ser executados antes que este caso de uso se inicie.
- **Atores envolvidos:** administrador do sistema.
- **Fluxo principal:** este caso de uso se inicia após o usuário ter realizado login, consultado uma aplicação, um nível e uma missão no sistema. Na sequência, ele lista as metas cadastradas, o usuário seleciona a meta que deseja realizar o procedimento e depois seleciona a atividade desejada: alterar meta ou sair. Se a atividade selecionada for alterar meta o sub-fluxo "alterar meta" será executado. Se a atividade selecionada for sair, o caso de uso termina.
- **Sub-fluxos:** "alterar meta" - o sistema exibe uma janela para alteração dos dados: descrição e valor. O sistema salva os dados e o caso de uso é reiniciado.
- **Fluxos alternativos:** nenhum.

5.3.2.19 Caso de uso Excluir Meta

- **Descrição:** este caso de uso aborda o procedimento adotado para a exclusão de metas cadastradas em missões do sistema.
- **Pré-condições:** os casos de uso "fazer login", "consultar aplicação", "consultar nível" e "consultar missão" devem ser executados antes que este caso de uso se inicie.
- **Atores envolvidos:** administrador do sistema.
- **Fluxo principal:** este caso de uso se inicia após o usuário ter realizado login, consultado uma aplicação, um nível e uma missão no sistema. Na sequência, ele lista as metas cadastradas, o usuário seleciona a meta que deseja realizar o procedimento e depois seleciona a atividade desejada: excluir meta ou Sair. Se a atividade selecionada for excluir meta, o sub-fluxo "excluir meta" será executado. Se a atividade selecionada for sair, o caso de uso termina.

- **Sub-fluxos:** "excluir meta" - o sistema exibe o código e a descrição da meta para confirmação da exclusão. O sistema exclui a meta e o caso de uso reinicia.
- **Fluxos alternativos:** nenhum.

5.3.2.20 Caso de uso Consultar Meta

- **Descrição:** este caso de uso aborda o procedimento adotado para a consulta das metas cadastradas nas missões do sistema.
- **Pré-condições:** os casos de uso "fazer login", "consultar aplicação", "consultar nível" e "consultar missão" devem ser executados antes que este caso de uso se inicie.
- **Atores envolvidos:** administrador do sistema.
- **Fluxo principal:** este caso de uso se inicia após o usuário ter realizado login, consultada uma aplicação, um nível e uma missão no sistema. Na sequência, ele lista as metas cadastradas, o usuário seleciona a meta que deseja realizar o procedimento e depois seleciona a atividade desejada: "consultar meta" ou sair. Se a atividade selecionada for consultar meta, o sub-fluxo "consultar meta" será executado. Se a atividade selecionada for sair, o caso de uso termina.
- **Sub-fluxos:** "consultar meta" - o sistema exibe o código e a descrição da meta para confirmação da visualização. O sistema exibe as informações da meta e o caso de uso reinicia.
- **Fluxos alternativos:** nenhum.

5.3.2.21 Caso de uso Inserir Recompensa

- **Descrição:** este caso de uso aborda o procedimento adotado para o cadastro de recompensas em missões no sistema.
- **Pré-condições:** os casos de uso "fazer login", "consultar aplicação", "consultar nível" e "consultar missão" devem ser executados antes que este caso de uso se inicie.
- **Atores envolvidos:** administrador do sistema.
- **Fluxo principal:** este caso de uso se inicia após o usuário ter realizado login, consultada uma aplicação, um nível e uma missão no sistema. Na sequência, o usuário seleciona a atividade desejada: inserir recompensa ou sair. Se a

atividade selecionada for inserir recompensa, o sub-fluxo "inserir recompensa" será executado. Se a atividade selecionada for sair, o caso de uso termina.

- **Sub-fluxos:** "inserir recompensa" - o sistema exibe uma janela para entrada dos seguintes dados: descrição e endereço da imagem. O sistema salva os dados e o caso de uso é reiniciado.
- **Fluxos alternativos:** nenhum.

5.3.2.22 Caso de uso Alterar Recompensa

- **Descrição:** este caso de uso aborda o procedimento adotado para a alteração de uma recompensa cadastrada em uma missão no sistema.
- **Pré-condições:** os casos de uso "fazer login", "consultar aplicação", "consultar nível" e "consultar missão" devem ser executados antes que este caso de uso se inicie.
- **Atores envolvidos:** administrador do sistema.
- **Fluxo principal:** este caso de uso se inicia após o usuário ter realizado login, consultada uma aplicação, um nível e uma missão no sistema. Na sequência, ele lista as recompensas cadastradas, o usuário seleciona a recompensa que deseja realizar o procedimento e depois seleciona a atividade desejada: alterar recompensa ou sair. Se a atividade selecionada for alterar recompensa o sub-fluxo "alterar recompensa" será executado. Se a atividade selecionada for sair, o caso de uso termina.
- **Sub-fluxos:** "alterar recompensa" - o sistema exibe uma janela para alteração dos dados: descrição e endereço da imagem. O sistema salva os dados e o caso de uso é reiniciado.
- **Fluxos alternativos:** nenhum.

5.3.2.23 Caso de uso Excluir Recompensa

- **Descrição:** este caso de uso aborda o procedimento adotado para a exclusão de recompensas cadastradas em uma missão do sistema.
- **Pré-condições:** os casos de uso "fazer login", "consultar aplicação", "consultar nível" e "consultar missão" devem ser executados antes que este caso de uso se inicie.

- **Atores envolvidos:** administrador do sistema.
- **Fluxo principal:** este caso de uso se inicia após o usuário ter realizado login, consultada uma aplicação, um nível e uma missão no sistema. Na sequência, ele lista as recompensas cadastradas, o usuário seleciona a recompensa que deseja realizar o procedimento e depois seleciona a atividade desejada: excluir recompensa ou Sair. Se a atividade selecionada for excluir recompensa, o sub-fluxo "excluir recompensa" será executado. Se a atividade selecionada for sair, o caso de uso termina.
- **Sub-fluxos:** "excluir recompensa" - o sistema exibe o código e a descrição da recompensa para confirmação da exclusão. O sistema exclui a recompensa e o caso de uso reinicia.
- **Fluxos alternativos:** nenhum.

5.3.2.24 Caso de uso Consultar Recompensa

- **Descrição:** este caso de uso aborda o procedimento adotado para a consulta das recompensas cadastradas em uma missão no sistema.
- **Pré-condições:** os casos de uso "fazer login", "consultar aplicação", "consultar nível" e "consultar missão" devem ser executados antes que este caso de uso se inicie.
- **Atores envolvidos:** administrador do sistema.
- **Fluxo principal:** este caso de uso se inicia após o usuário ter realizado login, consultada uma aplicação, um nível e uma missão no sistema. Na sequência, ele lista as recompensas cadastradas, o usuário seleciona a recompensa que deseja realizar o procedimento e depois seleciona a atividade desejada: consultar recompensa ou Sair. Se a atividade selecionada for consultar recompensa, o sub-fluxo "consultar recompensa" será executado. Se a atividade selecionada for Sair o caso de uso termina.
- **Sub-fluxos:** "consultar recompensa" - o sistema exibe o código e a descrição da recompensa para confirmação da visualização. O sistema exibe as informações da recompensa e o caso de uso reinicia.
- **Fluxos alternativos:** nenhum.

5.3.3 Requisitos Não Funcionais

Este tópico especifica todos os requisitos não funcionais do módulo Web Administrador, divididos em requisitos de usabilidade, documentação do usuário, comunicação, segurança e disponibilidade.

5.3.3.1 Usabilidade

O sistema deve prover uma interface simples, de fácil navegação para facilitar ao máximo o uso do mesmo por parte dos usuários.

5.3.3.2 Documentação do Usuário

O sistema deverá fornecer um manual de ajuda para a utilização e instalação das funções do sistema.

5.3.3.3 Comunicação

O sistema deve testar a comunicação com todos os meios envolvidos no processo, de forma que, se um deles falhar, o sistema deverá informar de imediato. Os meios envolvidos são: e-mail dos usuários cadastrados no módulo Web do Administrador.

5.3.3.4 Segurança

O sistema deve prover os mecanismos de segurança necessários para a autenticação de usuários para acessar os dados e funcionalidades do sistema.

5.3.3.5 Disponibilidade

O módulo Web do Administrador deverá estar disponível 24 horas por dia e sete dias por semana, executando-se em tempos pré-agendados pelos administradores para manutenção do sistema.

5.3.4 Diagrama de Atividades

Em seguida na Figura 35 abaixo, se encontra o diagrama de Atividades para o módulo Web do Administrador, o diagrama representa de uma forma genérica o processo de cadastro de Usuário, Aplicação, Nível, Missão, Meta e Recompensa do sistema.

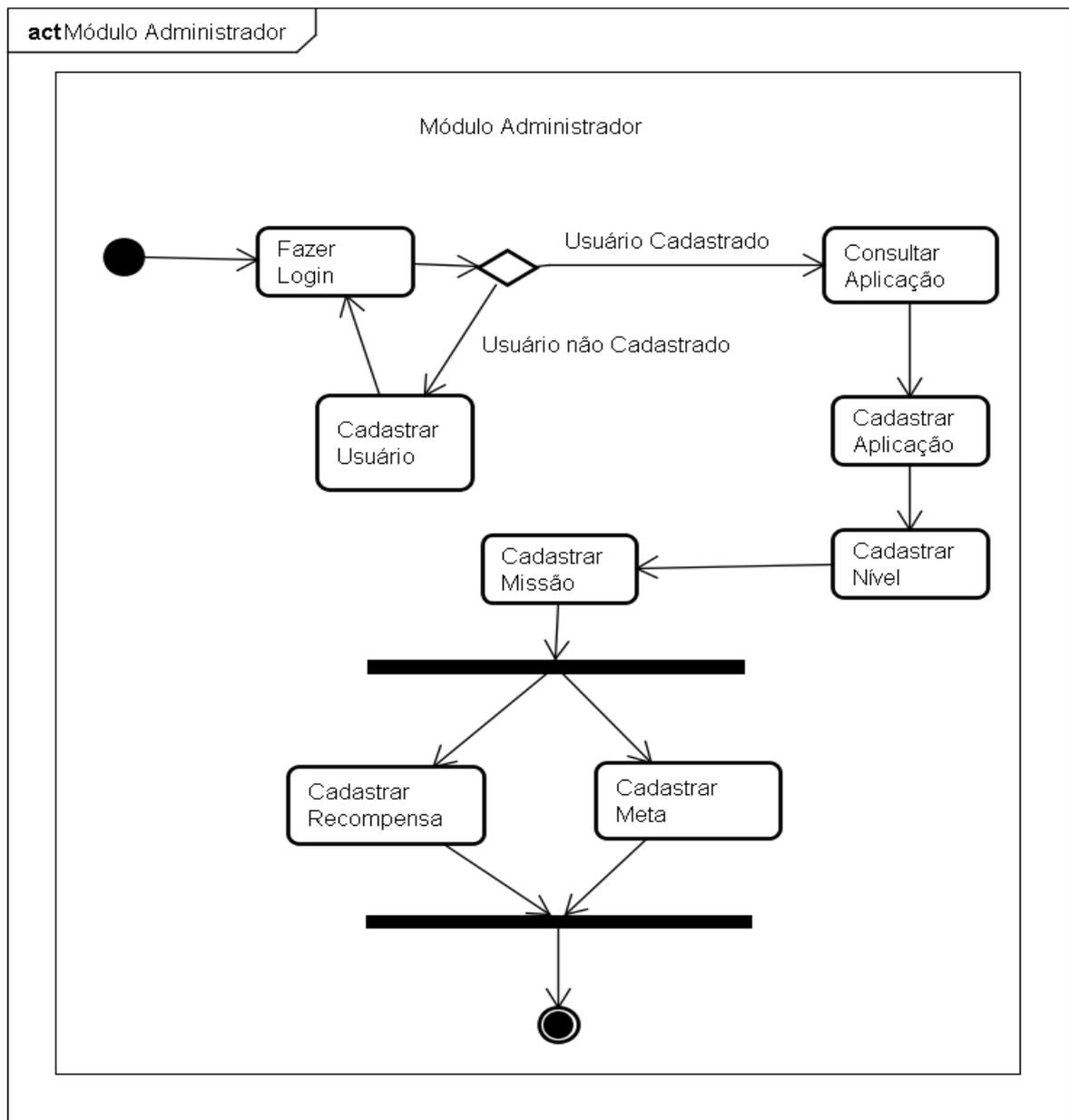


Figura 35: Diagrama de Atividades Módulo Administrador

Fonte: Elaboração própria do autor

5.4 Módulo da API Gamificada

A API Gamificada é a interface que disponibilizará os serviços de Gamificação, a partir dos dados cadastrados no módulo Web do Administrador para as aplicações clientes.

5.4.1 Casos de Uso

A Figura 36 apresenta o diagrama de caso de uso principal elaborado na modelagem do módulo da API Gamificada. Este diagrama apresenta todas as funcionalidades da API.

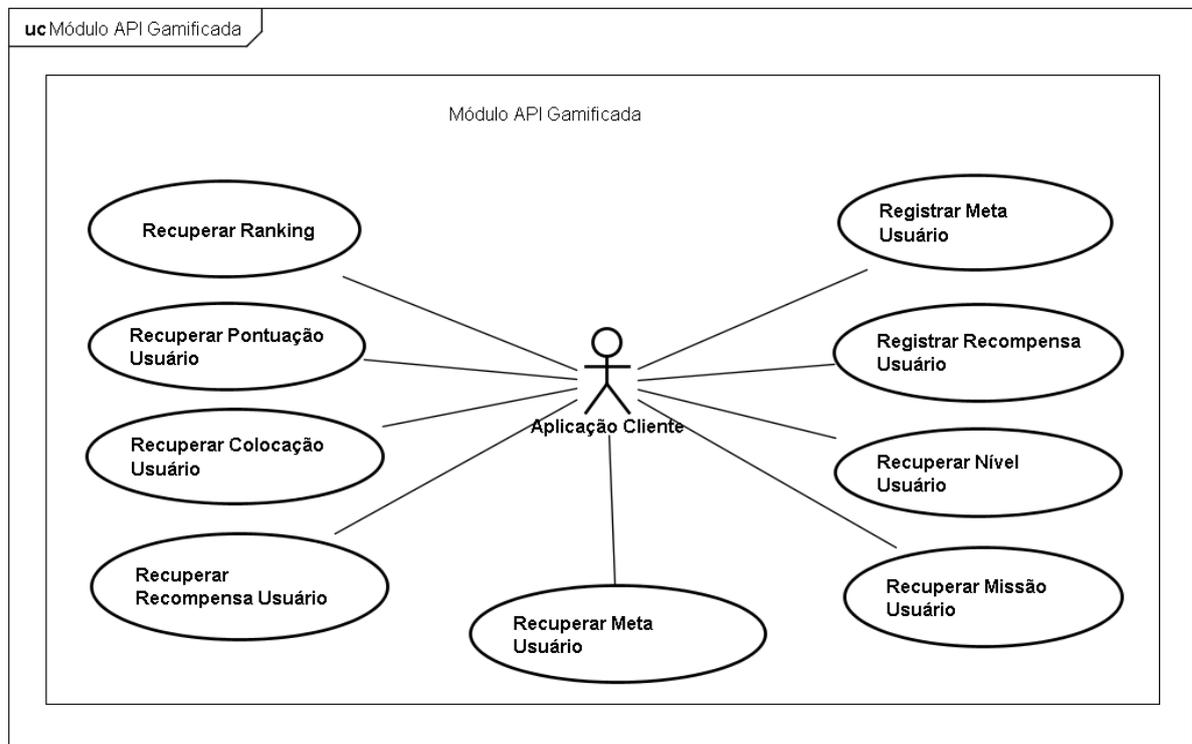


Figura 36: Casos de usos da API Gamificada

Fonte: Elaboração própria do autor

5.4.2 Requisitos Funcionais

Abaixo estão apresentados os fluxos de eventos para cada caso de uso supracitado acima.

5.4.2.1 Caso de uso Recuperar Ranking

- **Descrição:** este caso de uso aborda o procedimento adotado para recuperar o Ranking (classificação ordenada de acordo com determinados critérios) de usuários das aplicações clientes através da API Gamificada. O caso de uso se inicia quando a aplicação do cliente deseja recuperar Ranking dos usuários da API Gamificada e finaliza quando a consulta é concluída ou quando a aplicação do cliente cancela o procedimento.

- **Pré-condições:** a aplicação que irá estabelecer comunicação com a API deverá está cadastrada no módulo Web do Administrador.
- **Atores envolvidos:** aplicação cliente e API.
- **Fluxo principal:** este caso de uso se inicia após a aplicação do cliente ter realizado uma chamada ao método na API Gamificada. Ao selecionar a atividade recuperar ranking, o sub-fluxo "recuperar ranking" será executado.
- **Sub-fluxos:** "recuperar ranking" - a API Gamificada disponibiliza um método de conexão com a aplicação do cliente que deseja consultar. O método da API Gamificada necessita dos parâmetros de sua configuração: *código da aplicação, código do usuário, código do nível, código da missão, data do inicio e data do fim* da consulta. A consulta é realizada, o ranking é retornado e o caso de uso é reiniciado.
- **Fluxos alternativos:** houve uma tentativa de consulta de um ranking de usuários para uma aplicação que não possui usuários ou aplicações cadastradas e em seguida o caso de uso termina.

5.4.2.2 Caso de uso Recuperar Pontuação Usuário

- **Descrição:** este caso de uso aborda o procedimento adotado para recuperar a pontuação de usuários da API Gamificada. O caso de uso se inicia quando a aplicação do cliente deseja recuperar a pontuação de usuários da API Gamificada e finaliza quando a consulta é concluída ou quando a aplicação do cliente cancela o procedimento.
- **Pré-condições:** a aplicação que irá estabelecer comunicação com a API deverá está cadastrada no módulo Web do Administrador.
- **Atores envolvidos:** aplicação cliente e API.
- **Fluxo principal:** este caso de uso se inicia após a aplicação do cliente ter realizado uma chamada ao método na API Gamificada. Ao selecionar a atividade recuperar pontuação de usuário, o sub-fluxo "recuperar pontuação usuário" será executado.
- **Sub-fluxos:** "recuperar pontuação usuário" - a API Gamificada disponibiliza um método de conexão com a aplicação do cliente que deseja consultar. O método da API Gamificada necessita dos parâmetros para sua configuração: *código da aplicação, código do usuário, código do nível, código da missão, data do inicio e*

data do fim da consulta. A consulta é realizada, a pontuação do usuário é retornada e o caso de uso é reiniciado.

- **Fluxos alternativos:** houve uma tentativa de recuperar pontuação usuário que não possui usuário ou aplicação cadastrado e em seguida o caso de uso termina.

5.4.2.3 Caso de uso Recuperar Colocação Usuário

- **Descrição:** este caso de uso aborda o procedimento adotado para recuperar a colocação de usuários da API Gamificada. O caso de uso se inicia quando a aplicação do cliente deseja recuperar a colocação de um usuário da API Gamificada e finaliza quando a consulta é concluída ou quando a aplicação do cliente cancela o procedimento.
- **Pré-condições:** a aplicação que irá estabelecer comunicação com a API deverá está cadastrada.
- **Atores envolvidos:** aplicação cliente e API.
- **Fluxo principal:** este caso de uso se inicia após a aplicação do cliente ter realizado uma chamada ao método na API Gamificada. Ao selecionar a atividade recuperar colocação usuário, o sub-fluxo "recuperar colocação usuário" será executado.
- **Sub-fluxos:** "recuperar colocação usuário" - a API Gamificada disponibiliza um método de conexão com a aplicação do cliente que deseja consultar. O método da API Gamificada necessita dos parâmetros para sua configuração: *código da aplicação, código do usuário, código do nível, código da missão, data do inicio e data do fim* da consulta. A consulta é realizada, a colocação do usuário é retornada e o caso de uso é reiniciado.
- **Fluxos alternativos:** houve uma tentativa de recuperar a colocação de um usuário ou aplicação que não possui cadastrado no módulo Web do Administrador e em seguida o caso de uso termina.

5.4.2.4 Caso de uso Recuperar Recompensa Usuário

- **Descrição:** este caso de uso aborda o procedimento adotado para recuperar as recompensas dos usuários da API Gamificada. O caso de uso se inicia quando a aplicação do cliente deseja recuperar as recompensas de um usuário da API

Gamificada e finaliza quando a consulta é concluída ou quando a aplicação do cliente cancela o procedimento.

- **Pré-condições:** a aplicação que irá estabelecer comunicação com a API deverá está cadastrada no módulo Web do Administrador.
- **Atores envolvidos:** aplicação cliente e API.
- **Fluxo principal:** este caso de uso se inicia após a aplicação do cliente ter realizado uma chamada ao método na API Gamificada. Ao selecionar a atividade recuperar recompensa usuário, o sub-fluxo "recuperar recompensa usuário" será executado.
- **Sub-fluxos:** "recuperar recompensa usuário" - a API Gamificada disponibiliza um método de conexão com a aplicação do cliente que deseja consultar. O método da API Gamificada necessita dos parâmetros para sua configuração: *código da aplicação, código do usuário, código do nível, código da missão, data do inicio e data do fim* da consulta. A consulta é realizada e a recompensa do usuário é retornada e o caso de uso é reiniciado.
- **Fluxos alternativos:** houve uma tentativa de recuperar a recompensa de um usuário ou aplicação que não possui cadastrado na API Gamificada e em seguida o caso de uso termina.

5.4.2.5 Caso de uso Recuperar Meta Usuário

- **Descrição:** este caso de uso aborda o procedimento adotado para recuperar as metas dos usuários da API Gamificada. O caso de uso se inicia quando a aplicação do cliente deseja recuperar as metas de um usuário da API Gamificada e finaliza quando a consulta é concluída ou quando a aplicação do cliente cancela o procedimento.
- **Pré-condições:** a aplicação que irá estabelecer comunicação com a API deverá está cadastrada no módulo Web do Administrador.
- **Atores envolvidos:** aplicação cliente e API.
- **Fluxo principal:** este caso de uso se inicia após a aplicação do cliente ter realizado uma chamada ao método na API Gamificada. Ao selecionar a atividade recuperar meta usuário, o sub-fluxo "recuperar meta usuário" será executado.

- **Sub-fluxos:** "recuperar meta usuário" - a API Gamificada disponibiliza um método de conexão com a aplicação do cliente que deseja consultar. O método da API Gamificada necessita dos parâmetros para sua configuração: *código da aplicação, código do usuário, código do nível, código da missão, data do início e data do fim* da consulta. A consulta é realizada e a meta do usuário é retornada e o caso de uso é reiniciado.
- **Fluxos alternativos:** houve uma tentativa de recuperar a meta de um usuário ou aplicação que não possui cadastrado na API Gamificada e em seguida o caso de uso termina.

5.4.2.6 Caso de uso Recuperar Missão Usuário

- **Descrição:** este caso de uso aborda o procedimento adotado para recuperar as missões dos usuários da API Gamificada. O caso de uso se inicia quando a aplicação do cliente deseja recuperar as missões de um usuário da API Gamificada e finaliza quando a consulta é concluída ou quando a aplicação do cliente cancela o procedimento.
- **Pré-condições:** a aplicação que irá estabelecer comunicação com a API deverá está cadastrada no módulo Web do Administrador.
- **Atores envolvidos:** aplicação cliente e API.
- **Fluxo principal:** este caso de uso se inicia após a aplicação do cliente ter realizado uma chamada ao método na API Gamificada. Ao selecionar a atividade recuperar missão usuário, o sub-fluxo "recuperar missão usuário" será executado.
- **Sub-fluxos:** "recuperar missão usuário" - a API Gamificada disponibiliza um método de conexão com a aplicação do cliente que deseja consultar. O método da API Gamificada necessita dos parâmetros para sua configuração: *código da aplicação, código do usuário, código do nível, código da missão, data do início e data do fim* da consulta. A consulta é realizada e a missão do usuário é retornada e o caso de uso é reiniciado.
- **Fluxos alternativos:** houve uma tentativa de recuperar a missão de um usuário ou aplicação que não possui cadastro na API Gamificada, e em seguida o caso de uso termina.

5.4.2.7 Caso de uso Recuperar Nível Usuário

- **Descrição:** este caso de uso aborda o procedimento adotado para recuperar os níveis dos usuários da API Gamificada. O caso de uso se inicia quando a aplicação do cliente deseja recuperar os níveis de um usuário da API Gamificada e finaliza quando a consulta é concluída ou quando a aplicação do cliente cancela o procedimento.
- **Pré-condições:** a aplicação que irá estabelecer comunicação com a API deverá estar cadastrada no módulo Web do Administrador.
- **Atores envolvidos:** aplicação cliente e API.
- **Fluxo principal:** este caso de uso se inicia após a aplicação do cliente ter realizado uma chamada ao método na API Gamificada. Ao selecionar a atividade recuperar nível usuário, o sub-fluxo "recuperar nível usuário" será executado.
- **Sub-fluxos:** "recuperar nível usuário" - a API Gamificada disponibiliza um método de conexão com a aplicação do cliente que deseja consultar. O método da API Gamificada necessita dos parâmetros para sua configuração: *código da aplicação, código do usuário, código do nível, código da missão, data do início e data do fim* da consulta. A consulta é realizada e o nível do usuário é retornado e o caso de uso é reiniciado.
- **Fluxos alternativos:** houve uma tentativa de recuperar o nível de um usuário ou aplicação que não possui cadastro na API Gamificada e em seguida o caso de uso termina.

5.4.2.8 Caso de uso Registrar Recompensa Usuário

- **Descrição:** este caso de uso aborda o procedimento adotado para registrar as recompensas dos usuários da API Gamificada. O caso de uso se inicia quando a aplicação do cliente deseja inserir as recompensas de um usuário da API Gamificada e finaliza quando a inserção é concluída ou quando a aplicação do cliente cancela o procedimento.
- **Pré-condições:** a aplicação que irá estabelecer comunicação com a API deverá estar cadastrada no módulo Web do Administrador.
- **Atores envolvidos:** aplicação cliente e API.
- **Fluxo principal:** este caso de uso se inicia após a aplicação do cliente ter realizado uma chamada ao método na API Gamificada. Ao selecionar a

atividade registrar recompensa usuário, o sub-fluxo "registrar recompensa usuário" será executado.

- **Sub-fluxos:** "registrar recompensa usuário" - a API Gamificada disponibiliza um método de conexão com a aplicação do cliente que deseja utilizá-la. O método da API Gamificada necessita dos parâmetros para sua inserção: *código da aplicação, código do usuário, código do nível, código da missão e código da recompensa*. A inserção é realizada e o caso de uso é reiniciado.
- **Fluxos alternativos:** houve uma tentativa de inserir uma recompensa de um usuário ou aplicação que não possui cadastrado na API Gamificada, e em seguida o caso de uso termina.

5.4.2.9 Caso de uso Registrar Meta Usuário

- **Descrição:** este caso de uso aborda o procedimento adotado para registrar as metas dos usuários da API Gamificada. O caso de uso se inicia quando a aplicação do cliente deseja inserir as metas de um usuário da API Gamificada e finaliza quando a inserção é concluída ou quando a aplicação do cliente cancela o procedimento.
- **Pré-condições:** a aplicação que irá estabelecer comunicação com a API deverá está cadastrada no módulo Web do Administrador.
- **Atores envolvidos:** aplicação cliente e API.
- **Fluxo principal:** este caso de uso se inicia após a aplicação do cliente ter realizado uma chamada ao método na API Gamificada. Ao selecionar a atividade registrar meta usuário, o sub-fluxo "registrar meta usuário" será executado.
- **Sub-fluxos:** "registrar meta usuário" - a API Gamificada disponibiliza um método de conexão com a aplicação do cliente que deseja utilizá-la. O método da API Gamificada necessita dos parâmetros para sua inserção: *código da aplicação, código do usuário, código do nível, código da missão e código da meta*. A inserção é realizada e o caso de uso é reiniciado.
- **Fluxos alternativos:** houve uma tentativa de inserir uma meta de um usuário ou aplicação que não possui cadastrado na API Gamificada, e em seguida o caso de uso termina.

5.4.3 Requisitos Não Funcionais

Especificam todos os requisitos não funcionais da API Gamificada, divididos em requisitos de disponibilidade, manutenção, segurança e desempenho.

5.4.3.1 Disponibilidade

A API Gamificada deverá estar disponível 24 horas por dia e sete dias por semana, executando-se em tempos pré-agendados pelos administradores para manutenção do sistema.

5.4.3.2 Manutenção

A API Gamificada deverá manter seu padrão inicial, caso haja a necessidade de atualizações, deverá ser primeiramente feito um estudo de impacto nas aplicações dos clientes que fazem uso do seu serviço, levando em consideração que mudanças estruturais podem comprometer a comunicação da API Gamificada com as aplicações dos clientes.

5.4.3.3 Segurança

As aplicações dos clientes deverão estar previamente cadastradas no módulo Web do Administrador para terem acesso às informações contidas nas aplicações cadastradas.

5.4.3.4 Desempenho

A API Gamificada deve fornecer um desempenho aceitável pela aplicação do cliente, que seja rápida a invocação dos métodos que ela disponibiliza.

5.4.4 Diagrama de Atividades

Em seguida na Figura 37 abaixo, se encontra o Diagrama de Atividades para a API Gamificada. Ele foi elaborado de uma forma genérica, pois visa atender a todas as chamadas de métodos realizados entre a aplicação do cliente e a API gamificada.

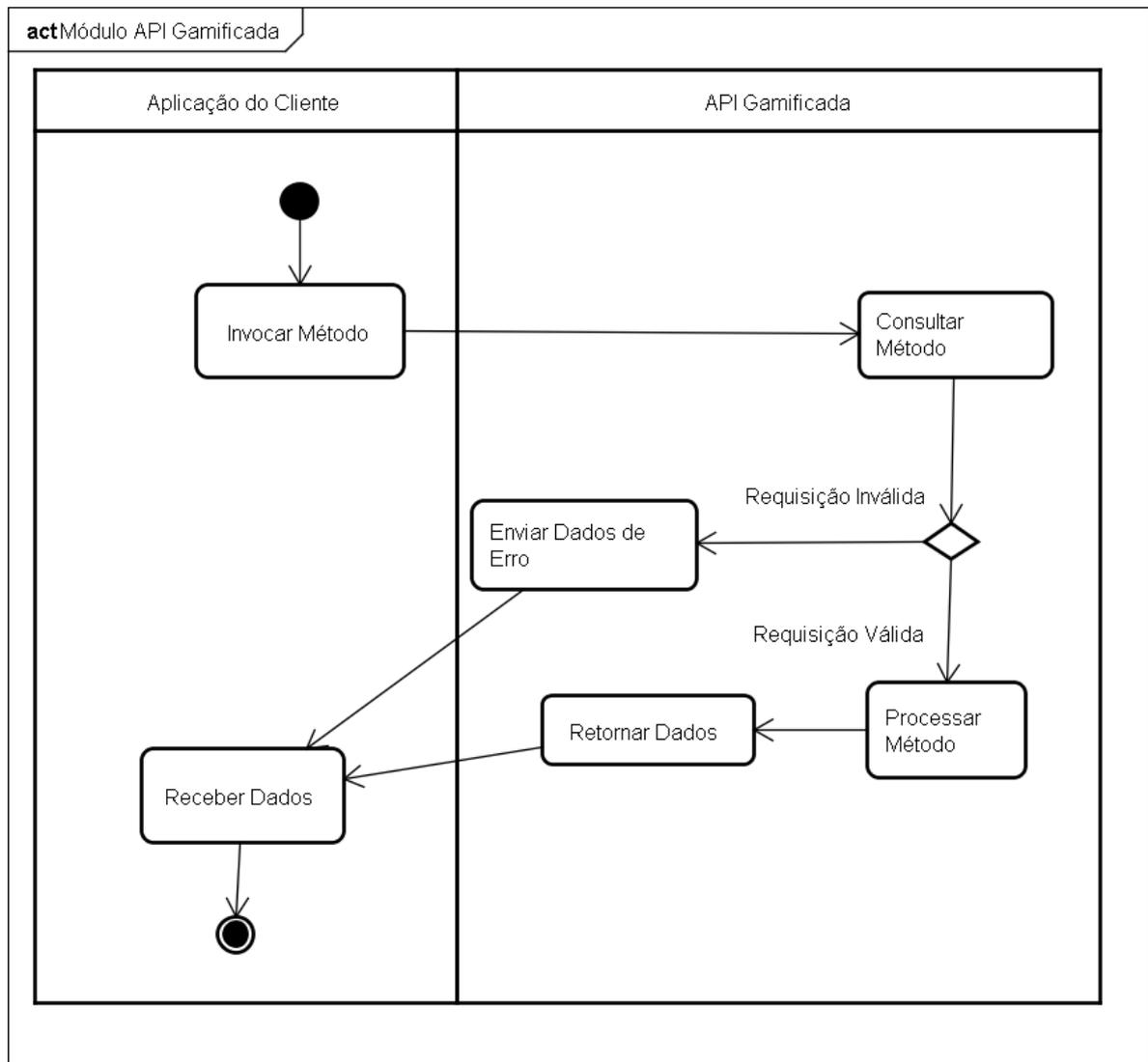


Figura 37: Diagrama de Atividades do Módulo API Gamificada.

Fonte: Elaboração própria do autor

5.5 Diagrama de Entidade Relacionamento

Ao se modelar uma base de dados é necessário que se tenha um grande entendimento da solução a ser proposta. A utilização de modelos é importante para mostrar uma visão clara e formal do problema modelado.

A modelagem da base de dados foi feita utilizando o software *DBDesigner*, que auxilia na modelagem de dados. A nomenclatura das tabelas segue o seguinte padrão:

- O nome da tabela é composto pela inicial *tbl* que é a abreviação para “tabela” e do nome da tabela, “tbl”+“nome_da_tabela”.

Ex: “tbl_aplicacao”

- O nome de um campo, que é chave primária, da tabela é composto pela inicial *pmk* que é a abreviação do termo *primary key*, com o tipo do campo e em seguida com o nome da tabela, “*pmk*”+“*tipo_do_campo*”+“*nome_da_tabela*”:

Ex: “*pmk_int_aplicacao*”.

- O nome de um campo, que é chave estrangeira, em uma tabela é composto pela inicial *fok* que é a abreviação do termo *foreign key*, com o tipo do campo e em seguida com o nome da tabela que lhe originou, “*fok*”+“*tipo_do_campo*”+“*nome_da_tabela_origem*”:

Ex: “*fok_int_administrador*”.

- O nome de um campo, que não é chave primária e também não é chave estrangeira, de uma tabela é composto pela abreviação das 3(três) primeiras letras do nome da tabela, “*abreviação*”+“*tipo_do_campo*”+“*nome_do_campo*”, usando uma tabela “*tbl_aplicacao*” e o campo título como exemplos, o campo fica da seguinte forma:

Ex: “*apl_str_titulo*”.

- O nome de um campo, que não é chave primária, também não é chave estrangeira e pertence a uma tabela criada pelo relacionamento de outras duas tabelas é composto pela abreviação das 3 (três) primeiras letras dos nomes de cada uma das duas tabelas consecutivamente, “*abreviação_primeira_tabela*” + “*abreviação_segunda_tabela*” + “*tipo_do_campo*” + “*nome_do_campo*”, usando uma tabela “*tbl_aplicacao_usuario*” e o campo título como exemplos, o campo fica da seguinte forma:

Ex: “*apl_usu_str_titulo*”.

- Para o padrão “*tipo_de_campo*” utilizam-se os seguintes formatos:
 - Tipo *inteiro*: “*int*”;
 - Tipo *texto*: “*str*”;

- Tipo *data*: “*dta*”.

As tabelas que compõem o sistema são listadas abaixo com suas respectivas descrições, de acordo com o padrão supracitado acima:

- *tbl_usuario_administrador*: armazena a identificação e dados básicos do usuário administrador da API Gamificada;
- *tbl_aplicacao*: armazena as informações da aplicações clientes cadastradas pelo administrador;
- *tbl_nivel*: armazena a quantidade de níveis que cada aplicação cliente possui;
- *tbl_recompensa*: armazena as recompensas que a API Gamificada possui;
- *tbl_missao*: armazena a quantidade de missões que cada nível possui;
- *tbl_missao_recompensa*: relaciona as missões cadastradas para cada uma das recompensas da API Gamificada;
- *tbl_meta*: armazena as metas que a API Gamificada possui;
- *tbl_missao_meta*: relaciona as missões cadastradas para cada uma das metas da API Gamificada;
- *tbl_recompensa_usuario*: armazena as recompensas para cada uma das missões que o usuário realizou;
- *tbl_meta_usuario*: armazena as metas alcançadas para cada uma das missões que o usuário realizou.

A Figura 38 abaixo apresenta a modelagem de dados realizada para a solução API Gamificada proposta.

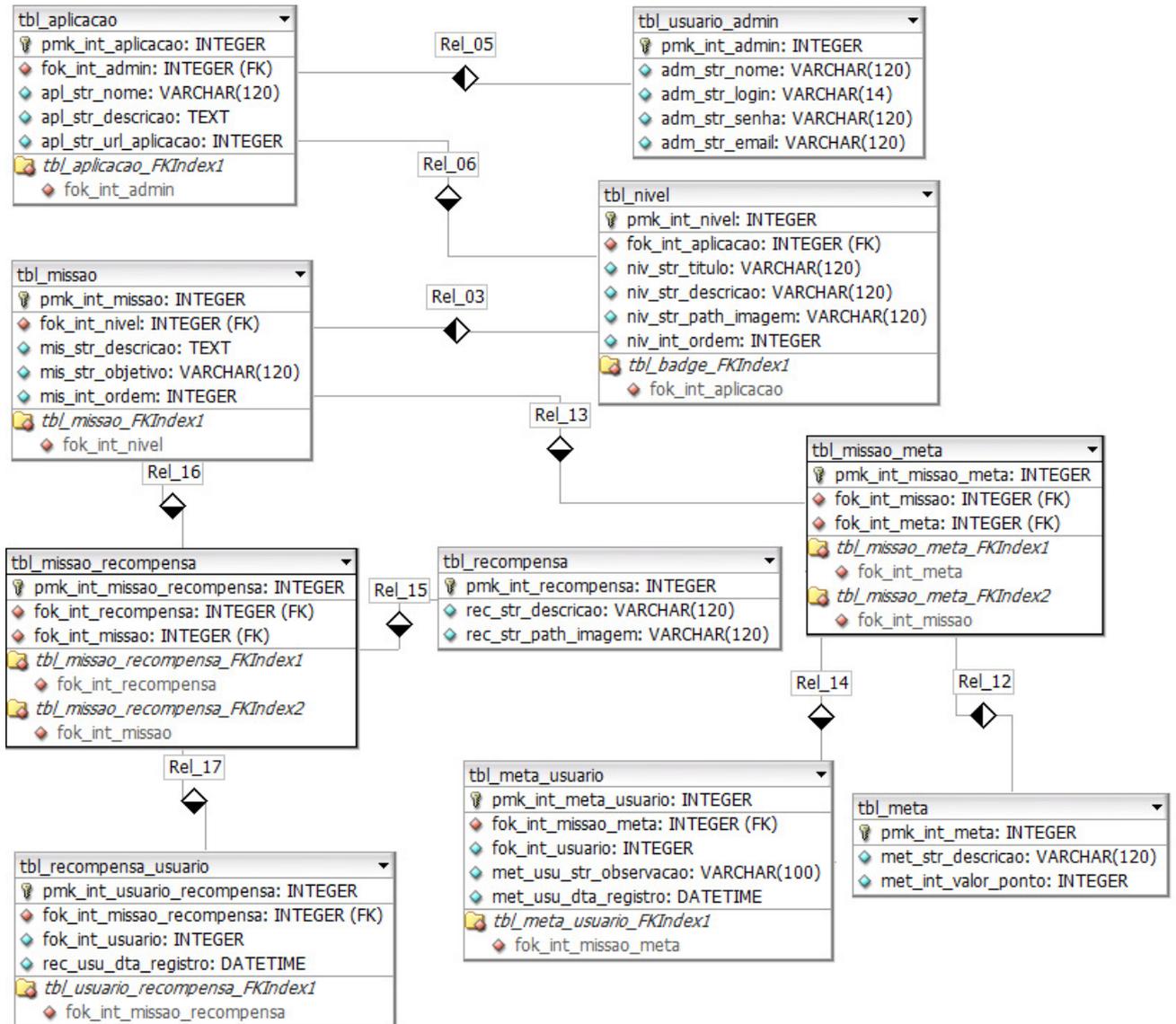


Figura 38 - Diagrama de Entidade e Relacionamento

Fonte: Elaboração própria do autor

5.5.1 Dicionário de Dados

No APÊNDICE I deste trabalho encontra-se o dicionário de dados gerado para a solução da API proposta.

6 CONCLUSÃO

A proposta de uma API Gamificada, que possa colaborar com o engajamento dos usuários, esteja sempre disponível e seja de fácil integração com aplicações clientes na Web visou contribuir com a concepção de soluções em ambientes gamificados. A Gamificação promove, segundo os autores estudados, além da disponibilidade dos usuários, uma forma de prover competitividade e retorno para as aplicações interessadas em manter um público ativo.

Deste modo, alcançou-se neste trabalho: a) o levantamento dos requisitos na literatura especializada e em aplicações gamificadas como o Duolingo, Livemocha e Foursquare; b) a análise dos requisitos para desenvolvimento da solução da API Gamificada; c) a documentação dos requisitos, tanto funcionais quanto não funcionais, para desenvolvimento da solução da API Gamificada; d) a modelagem dos diagramas de casos de uso, diagrama de classe, diagrama de atividades e diagrama de entidade e relacionamento, seguindo a UML.

Deste modo, acredita-se que a solução proposta possa ser eficiente por seguir conceitos da literatura especializada e aplicações gamificadas na Web. No entanto, uma das grandes dificuldades do processo de desenvolvimento de qualquer tipo de software é a certeza de que realmente o sistema atingiu a medida exata que se esperava.

Como trabalhos futuros, pretende-se implementar a API Gamificada modelada e fazer um estudo sobre qual é o seu impacto entre os usuários de aplicações clientes gamificadas na Web. Através destes dados, pretende-se aprimorar e alinhar os requisitos e as funcionalidades, desta proposta, buscando formas de incrementar cada vez mais a motivação e o engajamento dos usuários envolvidos nos processos de negócios inseridos nas diversas aplicações disponíveis na Web.

REFERÊNCIAS

- ANDER-EGG, Ezequiel. **Introducción a las técnicas de investigación social: para trabajadores sociales**. 7. ed. Buenos Aires: Humanitas, 1978, Terceira Parte APUD MARCONI, Marina de Andrade & LAKATOS, Eva Maria. **Técnicas de Pesquisa: planejamento e execução de pesquisa, elaboração, análise e interpretação de dados**. São Paulo: Atlas, 2006.
- BARTLE, R. A. **Who Plays MUAs?** Comms Plus!, October/November, 1990. Disponível em: <http://www.mud.co.uk/richard/hcde.htm>. Acesso em: 4 setembro de 2013.
- BLANCHETTE, Jasmin. **The Little Manual of API Design**. Trolltech, a Nokia company. 19 de junho de 2008.
- BOGOST, I. **Persuasive Games: The Expressive Power of Videogames**. 2007.
- BOOCH, G; RUMBAUGH, J e JACOBSON, I: **UML, Guia do Usuário: tradução**. Fábio Freitas da Silva, Rio de Janeiro, Campus , 2000.
- BOOCH, G; RUMBAUGH, J e JACOBSON, I: **UML, Guia do Usuário: tradução**; Fábio Freitas da Silva e Cristina de Amorin Machado, Rio de Janeiro, Campus , 2005.
- DETERDING, S.; SICART, M.; NACKE, L.; O'Hara, K.; Dixon, D. **Gamification. using game-design elements in non-gaming contexts**. In CHI '11.Vancouver, Canada. 2011.
- DUOLINGO. Disponível em: <http://www.duolingo.com>. Acesso em: 15 novembro. 2013.
- ELMASRI, Ramez; NAVATHE, Shamkant B., **Sistemas de Banco de Dados**. 6ª.ed.São Paulo: Addison Wesley, 2011.
- ELMASRI, Ramez; NAVATHE, Shamkant B.. **Sistemas de Banco de Dados**. 6ª edição. São Paulo: Addison Wesley, 2011.
- FOURSQUARE. Disponível em: <http://www.foursquare.com>. Acesso em: 17 novembro. 2013.
- GIL, A. C. **Métodos e técnicas da pesquisa social**. São Paulo: 1987.
- GUEDES, Gilleanes T. A. **UML 2 – Uma Abordagem Prática**. 2ª Edição, Novatec Editora, 2011.
- JACOBSON, D.; Brail, G.; Woods, D. **APIs: A Strategy Guide**. 1ª edição. O'Reilly Media, EUA, 2012.
- KAPP, Karl. **Gadgets, games, and gizmos for learning: tools and techniques for transferring know-how from boomers to gamers**. John Wiley and Sons. 2007.

KAPP, Karl. **The Gamification of Learning and Instruction: Game-Based Methods and Strategies for Training and Education**. San Francisco: Pfeiffer. ISBN: 978-1-118-09634-5. 336 pages. 2012.

KICKMEIER-RUST, M., PEIRCE, N., CONLAN, O., SCHWARZ, D., VERPOORTEN, D. & ALBERT, D. **Immersive Digital Games: The Interfaces for Next Generation E-Learning?**, Universal. 2007.

LIU, Yefeng; ALEXANDROVA, Todorka; NAKAJIMA, Tatsuo. (2011) Gamifying intelligent environments. Proceedings of the 2011 international ACM workshop on Ubiquitous meta user interfaces, Scottsdale, Arizona, USA.

LIVEMOCHA. Disponível em: <http://www.livemocha.com>. Acesso em: 2 novembro. 2013.

MCGONIGAL, J. **Reality is Broken: Why Games make Us Better and how they can Change the World**. Penguin Pr. 2011.

PRENSKY, M. **Digital-Game Based Learning**. Paragon House Edition, Paragon House. 2007.

PROGRAMMABLE WEB. Disponível em: <http://www.programmableweb.com>. Acesso em: 14 outubro. 2013.

QUADROS, G. B. F. **Comunidades de Aprendizagem.com/Livemocha: um jeito social de aprender idiomas**. Dissertação (Mestrado em Letras) Programa de Pós-Graduação em Letras, Universidade Católica de Pelotas, 2011.

SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDARSHAN,S.. **Sistema de Banco de Dados**.3.ed.São Paulo: Makron Books, 1999.

SOMMERVILLE, Ian. **Engenharia de Software**. 9 ed. São Paulo: Pearson, 2011.

STANLEY. G. **Gamification and language teaching**. Acesso em 01 novembro de 2013. Disponível em: <http://tllg.wikispaces.com/file/view/Stanley_Gamification.pdf>.

VAN ECK, R. **Digital Game Based Learning: It's Not Just the Digital Native Who Are Restless**. Educause Review, 41, 16-30. 2006.

ZICHERMANN, Gabe. (2011) Gamification by Design. ISBN 1449397670. 150 pages. O'Reilly.

APÊNDICE I – DICIONÁRIO DE DADOS

tbl_usuario_admin	
Termo	Definição do Termo
<pmk_int_admin>	<*Sequência numérica identificadora do usuário administrador.*>
<adm_str_nome>	<*Nome do usuário.*>
<adm_str_login>	<*Login do usuário.*>
<adm_str_senha>	<*Senha do usuário.*>
<adm_str_email>	<*E-mail do usuário.*>

tbl_aplicacao	
Termo	Definição do Termo
<pmk_int_aplicacao>	<*Sequência numérica identificadora da aplicação.*>
<fok_int_adm>	<*Chave estrangeira com a sequência numérica identificadora do usuário administrador.*>
<apl_str_nome>	<*Nome da aplicação.*>
<apl_str_descricao>	<*Descrição da aplicação.*>
<apl_str_url_aplicacao>	<*URL de acesso via navegador da aplicação.*>

tbl_nivel	
Termo	Definição do Termo
<pmk_int_nivel>	<*Sequência numérica identificadora do nível.*>
<fok_int_aplicacao>	<*Chave estrangeira com a sequência numérica identificadora da aplicação.*>
<niv_str_titulo>	<*Título do nível.*>

<niv_str_descricao>	<*Descrição do nível.*>
<niv_str_path_imagem>	<*Endereço da imagem do nível.*>
<niv_int_ordem>	<*Ordem do nível.*>

tbl_missao	
Termo	Definição do Termo
<pmk_int_missao>	<*Sequência numérica identificadora da missão.*>
<fok_int_nivel>	<*Chave estrangeira com a sequência numérica identificadora do nível.*>
<mis_str_descricao>	<*Descrição da missão.*>
<mis_str_objetivo>	<*Descrição do objetivo da missão.*>
<mis_int_ordem>	<*Ordem da missão.*>

tbl_missao_recompensa	
Termo	Definição do Termo
<pmk_int_missao_recompensa>	<*Sequência numérica identificadora do relacionamento de cada missão com cada recompensa*>
<fok_int_recompensa>	<*Chave estrangeira com a sequência numérica identificadora da recompensa.*>
<fok_int_missao>	<*Chave estrangeira com a sequência numérica identificadora da missão.*>

tbl_missao_meta	
Termo	Definição do Termo
<pmk_int_missao_meta>	<*Sequência numérica identificadora do relacionamento de cada missão com cada meta.*>
<fok_int_meta>	<*Chave estrangeira com a sequência

	numérica identificadora da meta.*>
<fok_int_missao>	<*Chave estrangeira com a sequência numérica identificadora da missão.*>

tbl_recompensa_usuario	
Termo	Definição do Termo
<pmk_int_usuario_recompensa>	<*Sequência numérica identificadora do relacionamento de cada missão e recompensa com cada usuário.*>
<fok_int_missao_recompensa>	<*Chave estrangeira com a sequência numérica identificadora do relacionamento missão com a recompensa.*>
<fok_int_usuario>	<*Chave estrangeira com a sequência numérica identificadora do usuário.*>
<rec_usu_dta_registro>	<*Data de conquista da recompensa.*>

tbl_meta_usuario	
Termo	Definição do Termo
<pmk_int_meta_usuario>	<*Sequência numérica identificadora do relacionamento de cada missão e meta com cada usuário.*>
<fok_int_missao_meta>	<*Chave estrangeira com a sequência numérica identificadora do relacionamento missão com a meta.*>
<fok_int_usuario>	<*Chave estrangeira com a sequência numérica identificadora do usuário.*>
<met_usu_str_observacao>	<*Observação da meta conquistada.*>
<met_usu_dta_registro>	<*Data de conquista da meta.*>

tbl_recompensa	
Termo	Definição do Termo

<pmk_int_recompensa>	<*Sequência numérica identificadora da recompensa.*>
<rec_str_descricao>	<*Descrição da recompensa.*>
<rec_str_path_imagem>	<*Endereço da imagem da recompensa.*>

tbl_meta	
Termo	Definição do Termo
<pmk_int_meta>	<*Sequência numérica identificadora da meta.*>
<met_str_descricao>	<*Descrição da meta.*>
<met_int_valor_ponto>	<*Valor de pontuação da meta.*>