

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
CÂMPUS CORNÉLIO PROCÓPIO  
DIRETORIA DE GRADUAÇÃO E EDUCAÇÃO PROFISSIONAL  
DEPARTAMENTO DE COMPUTAÇÃO  
ENGENHARIA DE SOFTWARE**

**CAIO THIZIO, DAFNY GARCIA, RAFAEL SANTOS**

**MEMORY-CLOUD**

**DOCUMENTO DE PROJETO DE SOFTWARE**

**CORNÉLIO PROCÓPIO  
2018**

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>3</b>
1.1	CONTEXTO	3
1.2	JUSTIFICATIVA	3
1.3	PROPOSTA	4
1.4	ORGANIZAÇÃO DO DOCUMENTO	5
<b>2</b>	<b>DESCRIÇÃO GERAL DO SISTEMA</b>	<b>6</b>
2.1	OBJETIVOS(GERAIS E ESPECÍFICOS)	6
2.2	LIMITES E RESTRIÇÕES	6
2.3	DESCRIÇÃO DOS USUÁRIOS DO SISTEMA	6
<b>3</b>	<b>DESENVOLVIMENTO DO PROJETO</b>	<b>8</b>
3.1	TECNOLOGIAS E FERRAMENTAS	8
3.1.1	Linguagem de Programação	8
3.1.2	Sistema Gerenciador de Banco de Dados	8
3.1.3	Ferramentas	8
3.1.3.1	Desenvolvimento	8
3.1.3.2	Documentação	8
3.1.3.3	Gerenciamento do Projeto	8
3.2	METODOLOGIA DE DESENVOLVIMENTO(CICLO DE VIDA E EQUIPE)	9
3.3	CRONOGRAMA PREVISTO	10
<b>4</b>	<b>REQUISITOS DO SISTEMA</b>	<b>11</b>
4.1	REQUISITOS FUNCIONAIS	11
4.2	REQUISITOS NÃO-FUNCIONAIS	11
4.3	DIAGRAMAS DE CASOS DE USO	12
<b>5</b>	<b>ANÁLISE DO SISTEMA</b>	<b>13</b>
5.1	ARQUITETURA DO SISTEMA	13
5.1.1	Arquitetura cliente/servidor Multicamadas	13
5.1.2	Infraestrutura	14
5.2	MODELO DO BANCO DE DADOS	15
5.2.1	Modelo Lógico	15
5.2.2	Dicionário de Dados	15
5.2.2.1	Entidade: admin	15
5.2.2.2	Entidade: achievement	15
5.2.2.3	Entidade: game	16
5.2.2.4	Entidade: integerDict	16
5.2.2.5	Entidade: floatDict	16
5.2.2.6	Entidade: stringDict	16
5.2.2.7	Entidade: boolDict	16
5.2.2.8	Entidade: achievementDone	16
5.2.2.9	Entidade: user	17
5.3	DIAGRAMA DE CLASSES	18
5.4	DIAGRAMA DE ATIVIDADES	19
5.4.1	Login	19
5.4.2	Novo Jogo	20
5.4.3	Gerenciar Achievements	21
5.5	DIAGRAMA DE SEQUENCIA	22

5.5.1	Requisição	22
<b>6</b>	<b>IMPLEMENTAÇÃO</b>	<b>23</b>
6.1	PROTÓTIPO DE TELAS	23
6.1.1	Unity	23
6.1.1.1	Menu	23
6.1.1.2	Settings	23
6.1.2	Dashboard	24
6.1.2.1	Login	24
6.1.2.2	SignUp	24
6.1.2.3	Games	25
6.1.2.4	Game	25
6.1.2.5	Achievements	26
6.2	DESCRIÇÃO DO CÓDIGO	27
6.2.1	Diagrama de Implementação	27
6.2.2	Esquema GraphQL	28
6.2.2.1	Query	28
6.2.2.2	Mutation	28
<b>7</b>	<b>CONSIDERAÇÕES FINAIS</b>	<b>29</b>
	<b>REFERÊNCIAS</b>	<b>30</b>

# 1 INTRODUÇÃO

## 1.1 CONTEXTO

Os primeiros jogos online comerciais começaram a aparecer em 1989 e eram jogados via ligação direta local ou direta internacional. Os jogadores conectavam entre si ligando para casa de seus "adversários" através do modem de seus computadores e a linha telefônica convencional. A desvantagem nessa época era jogar somente com um adversário e tendo que conhecê-lo de sua cidade ou ser parentesco (se não quisesse pagar uma conta telefônica exorbitante).

O Brasil é o maior consumidor de games na América Latina. Embora esteja na liderança regional, esse é um mercado que está apenas no início. Muitas áreas ainda sofrem com a falta de internet banda larga e do acesso a computadores e consoles mais "turbinados". Além disso, esse mercado está também invadindo as empresas, que estão utilizando a gamificação de diversas formas, sendo os treinamentos a principal delas.

O investimento em marketing de grandes marcas, como Sony e Microsoft, e os smartphones com cada vez mais capacidade de processamento só tende a aumentar a demanda. (SAGA, 2018)

## 1.2 JUSTIFICATIVA

Ferramentas para persistência de progressos de jogos Unity, no cenário atual, são pouco acessíveis para desenvolvedores. Serviços nesse segmento são caros, além de existirem poucas variedades. Nenhuma destas ferramentas possuem um código open source. Desenvolvedores de jogos de pequenas empresas, sofrem atualmente para que seus progressos de jogos possam ser salvos, enquanto trabalham em melhorias ou a continuidade de seus jogos.

As ferramentas gratuitas no mercado atualmente, trabalham com jogos na Steam ou com jogos completos, não sendo possível que o desenvolvedor teste e ainda assim mantenha seu progresso, sem a versão final.

Das ferramentas mais utilizadas, temos PlayFab e GameSparks. Para questão de comparação, usaremos a PlayFab. A ferramenta oferece:

- Multiplataforma de autenticação, para que os players possam jogar em vários dispositivos.
- Armazenamento de informações do jogador e estado do jogo na nuvem para que os dados nunca sejam perdidos.
- Compartilhamento de vários dispositivos e jogos.
- Compartilhamento com outros jogadores.
- Escolha do nível de acesso do cliente do jogo.
- Estatísticas do jogador para placares, matchmaking e segmentação. Suporte.

Para que as empresas possam utilizar esses serviços, o preço de início do pacote Profissional é de US\$299/mês, existem planos grátis, porém os mesmos podem sofrer modificações a qualquer momento:

*Our latest Native SDK release (2.3) will no longer support integration with iOS and going forward we will not be supporting or updating the iOS SDK.*  
(GOOGLE, 2017)

*The following nodes and edges will now only return empty data sets:*  
/achievement\_id  
/app/achievements  
/app/scores  
/user/achievements  
/user/invitable\_friends  
/user/scores  
/user/taggable\_friends  
(FACEBOOK, 2018)

Com o crescente mercado de jogos e interesse de novos desenvolvedores tanto em startups quanto individuais, fazer parte da comunidade desenvolvedora de jogos vem sendo um desafio. Ferramentas como PlayFab têm acesso limitado por possuírem código fechado, alta complexidade de desenvolvimento e configuração e assinaturas dispendiosas.

O projeto de uma ferramenta de persistência de jogos será de muita importância tanto para usuários de jogos Unity quanto para desenvolvedores. A API de outras ferramentas, além de muito custosas são muito complicadas. A API a ser desenvolvida é mais simplificada, pois possui conceitos similares à API da própria Unity, além da vantagem de custo. Com esta nova abordagem, usuários normais poderão interagir gratuitamente com amigos do Facebook através de rankings. Aos desenvolvedores, o benefício virá pela tentativa de trazer um tempo menor de desenvolvimento, tendo como um dos requisitos uma curva menor de aprendizado de uso da ferramenta.

### 1.3 PROPOSTA

O projeto será uma ferramenta de servidor que guardará o progresso de jogos Unity. O cliente estará usando o Facebook como identificador de usuários, criando rankings e cada jogo será salvo somente uma vez para que progresso atual possa prescrever o anterior. O usuário deverá logar primeiramente no facebook para que o jogo seja salvo.

Para o desenvolvedor, teremos uma API simplificada onde o usuário desenvolvedor irá configurar na própria plataforma Unity o cliente e poderá usar o servidor personalizado ou disponibilizado por padrão. O ranking que será mostrado ao cliente também poderá ser personalizado pelo desenvolvedor.

A implementação do projeto teria como primeiro impacto ser uma alternativa open source aos serviços pagos. Por ser de código livre, cada desenvolvedor poderá personalizar o back-end seguindo o gosto pessoal e estilo de programação, além de poder personalizar o próprio

servidor, de acordo com as preferências. Sobre finanças, reforça-se que é uma ferramenta open source e os usuários não precisariam pagar para fazer uso dela.

#### 1.4 ORGANIZAÇÃO DO DOCUMENTO

O documento está estruturado em tópicos, ou capítulos, onde cada tópico aborda uma ideia central e divide esta ideia em subtópicos.

O tópico seguinte aborda a Descrição Geral do Sistema, aonde serão apresentados objetivos gerais e específicos do sistema, bem como algumas de suas restrições e os usuários que poderão se beneficiar do produto.

O terceiro tópico trata do Desenvolvimento do Projeto, abordando algumas das tecnologias e ferramentas utilizadas para implementação do projeto; alguns dos métodos e processos utilizados para que o projeto se desenvolva, bem como os artefatos gerados durante os mesmos; traçar um plano de metas que ajudará na elaboração de um cronograma de desenvolvimento.

O quarto tópico trata dos requisitos envolvidos no projeto. Os requisitos serão levantados e classificados como funcionais ou não-funcionais e por meio de sua interpretação será possível a elaboração de diagramas de caso de uso para melhor visualização dos stakeholders e ações que os mesmos poderão performar no produto.

O quinto tópico trata da arquitetura do sistema. Analisar a arquitetura do software, fazer um mapeamento dos dados e como eles se distribuem no modelo lógico e apresentar diagramas de classes e de atividades.

O sexto tópico trata da implementação. Por meio de protótipos de tela e detalhamento da codificação, o documento retratará alguns aspectos do que foi desenvolvido no projeto.

O sétimo e último tópico discutirá os resultados da aplicação do produto, impactos e considerações finais.

## 2 DESCRIÇÃO GERAL DO SISTEMA

### 2.1 OBJETIVOS(GERAIS E ESPECÍFICOS)

Como foi descrito na seção anterior, um desenvolvedor de jogos Unity quando necessita de uma ferramenta de persistência de progresso online para algum de seus projetos, deve optar por restritos serviços que têm alto custo financeiro e possuem seu código fechado.

Como alternativa a esta situação, o objetivo geral deste projeto pode ser definido como: desenvolver uma solução open-source para o salvamento de progresso em jogos que utilizam a Engine Unity. Uma Engine nada mais é que uma biblioteca, um pacote de funcionalidades que são disponibilizadas para facilitar o desenvolvimento de um jogo e impedir que sua criação tenha seja feita do zero.

Por meio da delimitação do objetivo geral, é possível traçar alguns objetivos específicos que visam trazer mais portabilidade ao projeto:

- A ferramenta deve ter seu código aberto e totalmente personalizável, para que desenvolvedores possam utilizá-la da forma que melhor atender suas demandas;
- De forma a unificar um pouco do processo, usuários de jogos que utilizem a ferramenta deverão iniciar sessão na rede social Facebook para que o servidor colete as informações;
- Informações geradas pelos progressos dos jogos estarão disponíveis em um website para que desenvolvedores e usuários interajam com a ferramenta, por meio da criação de rankings.

### 2.2 LIMITES E RESTRIÇÕES

Alguns dos limites do projeto inicial contemplam:

- O sistema será monoplataforma para autenticação, utilizando apenas o Facebook para logar-se. Outras formas de autenticação podem ser implementadas posteriormente;
- O sistema restringe seu funcionamento a jogos que utilizam a engine Unity para salvamento de progressos;
- A única forma de interação entre usuários será, inicialmente, feita por meio dos rankings. Outras formas de interação deverão ser consideradas no futuro, como compartilhamento de informações e progressos entre usuários.

### 2.3 DESCRIÇÃO DOS USUÁRIOS DO SISTEMA

Os principais usuários finais do projeto deverão ser:

**Desenvolvedor** O desenvolvedor de jogos com a engine Unity será o principal usuário da ferramenta. Por meio dela, ele deverá ser capaz de implementar, dentro de seus projetos, formas para manter a persistência dos progressos realizados dentro dos jogos. Como a ferramenta possui código aberto, o desenvolvedor poderá adaptá-la da forma que melhor atender suas demandas. Ele poderá contar com bibliotecas predefinidas para evitar que o jogo perca seu progresso toda vez que seja reiniciado.

**Jogador** O jogador de jogos que utilizam a ferramenta para persistir as informações de progresso poderá acompanhar seus avanços por meio de um site, cuja autenticação se dá por meio de login pelo Facebook. No site será possível acompanhar um ranking com amigos que também joguem games que utilizam a ferramenta.



### 3 DESENVOLVIMENTO DO PROJETO

#### 3.1 TECNOLOGIAS E FERRAMENTAS

##### 3.1.1 Linguagem de Programação

**Servidor** NodeJS(ES8) e GraphQL

**Cliente** C# e GraphQL

##### 3.1.2 Sistema Gerenciador de Banco de Dados

MongoDB para persistência e Redis para cache.

##### 3.1.3 Ferramentas

###### 3.1.3.1 Desenvolvimento

**Servidor** NodeJS, Express, Mongoose, Kitematic, GraphQLApp

**Cliente** Unity3D 2017.0f3, JetBrains Rider, GraphQLApp

**Versionamento** GitHub

###### 3.1.3.2 Documentação

- Astah
- MySQL Workbench
- TeXstudio
- GitHub
- Google Drive
- Cloudcraft

###### 3.1.3.3 Gerenciamento do Projeto

- GitHub(boards, projects, issues, milestones)
- Discord

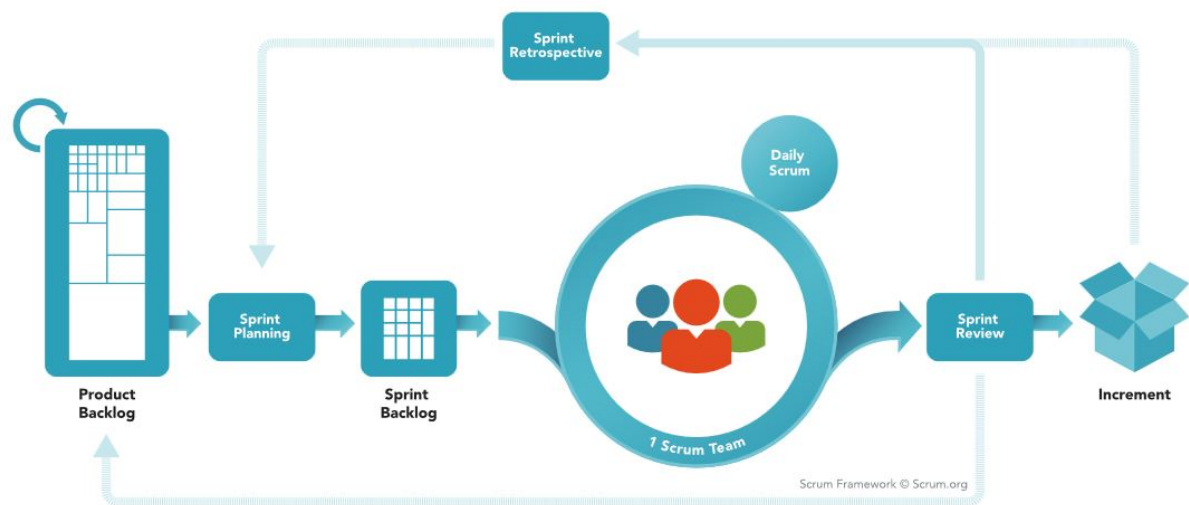
### 3.2 METODOLOGIA DE DESENVOLVIMENTO(CICLO DE VIDA E EQUIPE)

O desenvolvimento será iterativo e incremental a equipe é composta por:

**Product Owner** 2DverseStudio

**Development Team** Caio Thizio, Dafny Garcia, Rafael Santos

**Scrum Master** Rafael Santos



Fonte: (SCRUM, 2018)

**Product Backlog** funcionalidades ordenadas de acordo com sua importância.

**Sprint Planning** reunião para definir o Sprint Backlog.

**Sprint Backlog** lista de funcionalidades a serem implementadas no Sprint.

**Sprint** fase de implementação do que foi definido no Sprint Backlog, terá duração de uma semana.

**Sprint Review** reunião para mostrar o que foi feito no Sprint.

**Daily Scrum** reuniões diárias para deixar o time a par do desenvolvimento, realizadas via Discord.

**Sprint Retrospective** reunião de retrospectiva do Sprint, para definir o que pode ser melhorado no processo.

### 3.3 CRONOGRAMA PREVISTO

<b>Data</b>	<b>Tarefa</b>	<b>Responsáveis</b>
13/03	Concebimento da ideia	Caio, Dafny, Rafael
20/03	Coleta de requisitos	Rafael
27/03	Classificação de requisitos	Caio, Dafny
03/04	Início da redação do documento(capítulos 1, 2 e 3)	Dafny
10/04	Término da redação do documento(capítulos 1, 2 e 3) e redação	Caio, Dafny, Rafael
17/04	Início do desenvolvimento do projeto e documentação	Caio, Dafny, Rafael
24/04	Configuração inicial do servidor	Caio
01/05	Autenticação com Facebook(cliente e servidor)	Dafny
08/05	Salvamento e carregamento do progresso	Rafael
15/05	Quadro de liderança(ranking)	Caio, Dafny, Rafael

## 4 REQUISITOS DO SISTEMA

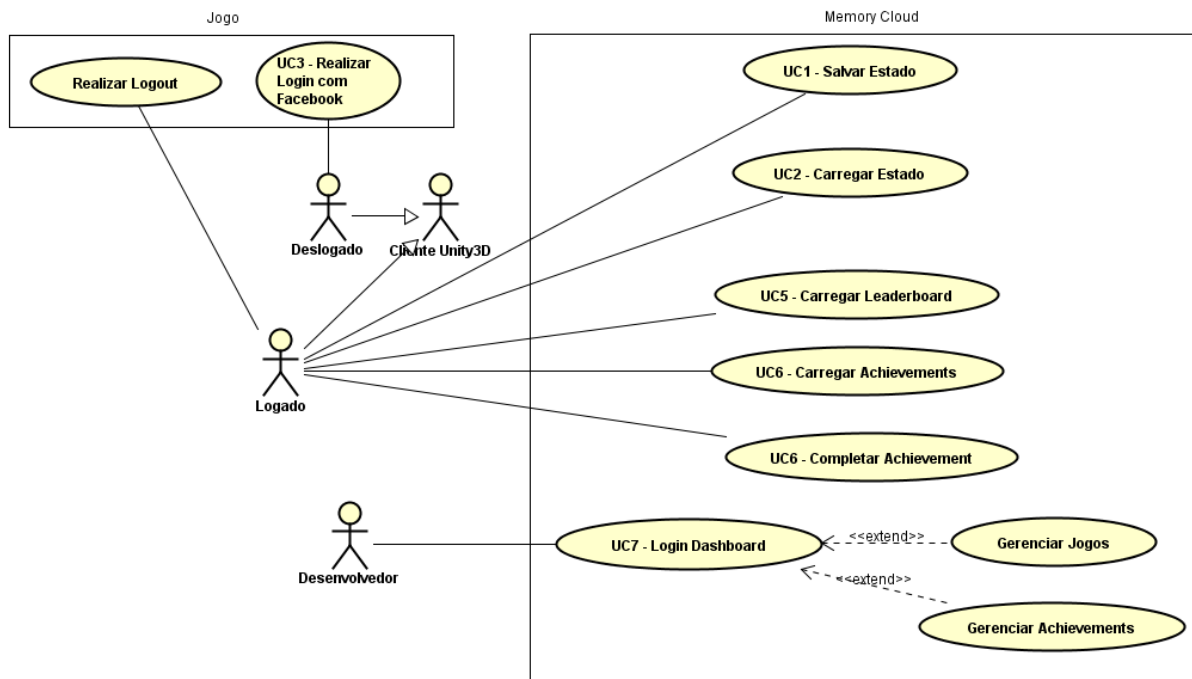
### 4.1 REQUISITOS FUNCIONAIS

ID	Funcionalidade	Prioridade
RF1	Salvar inteiros, floats, booleans e strings.	Essencial
RF2	Carregar inteiros, floats, booleans e strings.	Essencial
RF3	Usar o Facebook para realizar login.	Essencial
RF4	Suportar múltiplos jogos.	Essencial
RF5	Gerar Leaderboards a partir dos inteiros.	Importante
RF6	Sistema de achievements.	Desejável
RF7	Dashboard para configuração dos jogos.	Essencial

### 4.2 REQUISITOS NÃO-FUNCIONAIS

ID	Requisito	Categoria	Prioridade
RNF1	As senhas dos usuários do Dashboard devem ser criptografadas.	Segurança	Essencial
RNF2	Save deve ser feito em uma requisição ao servidor.	Usabilidade	Importante
RNF3	Load deve ser feito em uma requisição ao servidor.	Usabilidade	Importante
RNF4	Os Leaderboards devem ficar no cache por 120 segundos.	Usabilidade	Desejável
RNF5	O servidor deve utilizar o protocolo HTTPS.	Usabilidade	Essencial

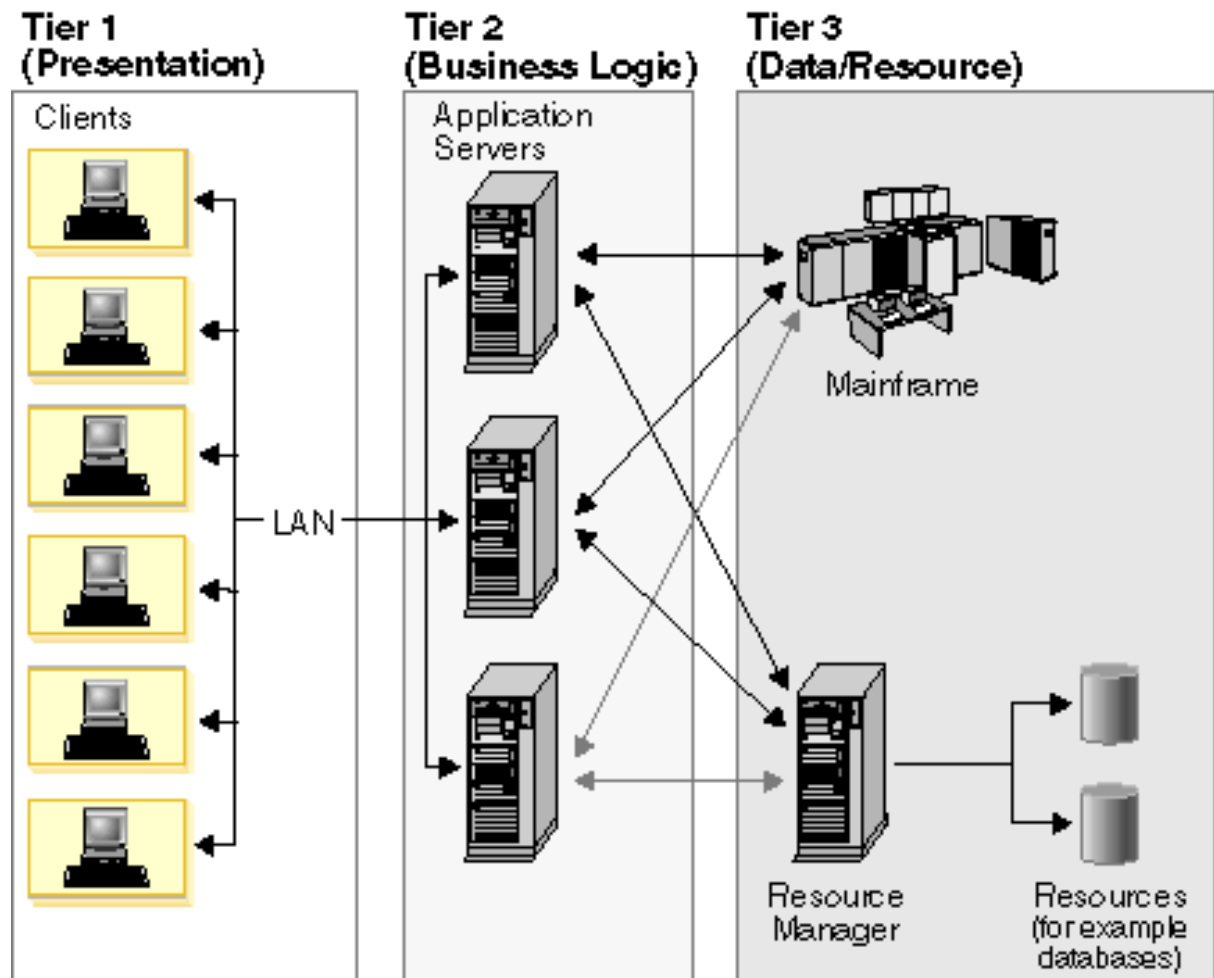
### 4.3 DIAGRAMAS DE CASOS DE USO



## 5 ANÁLISE DO SISTEMA

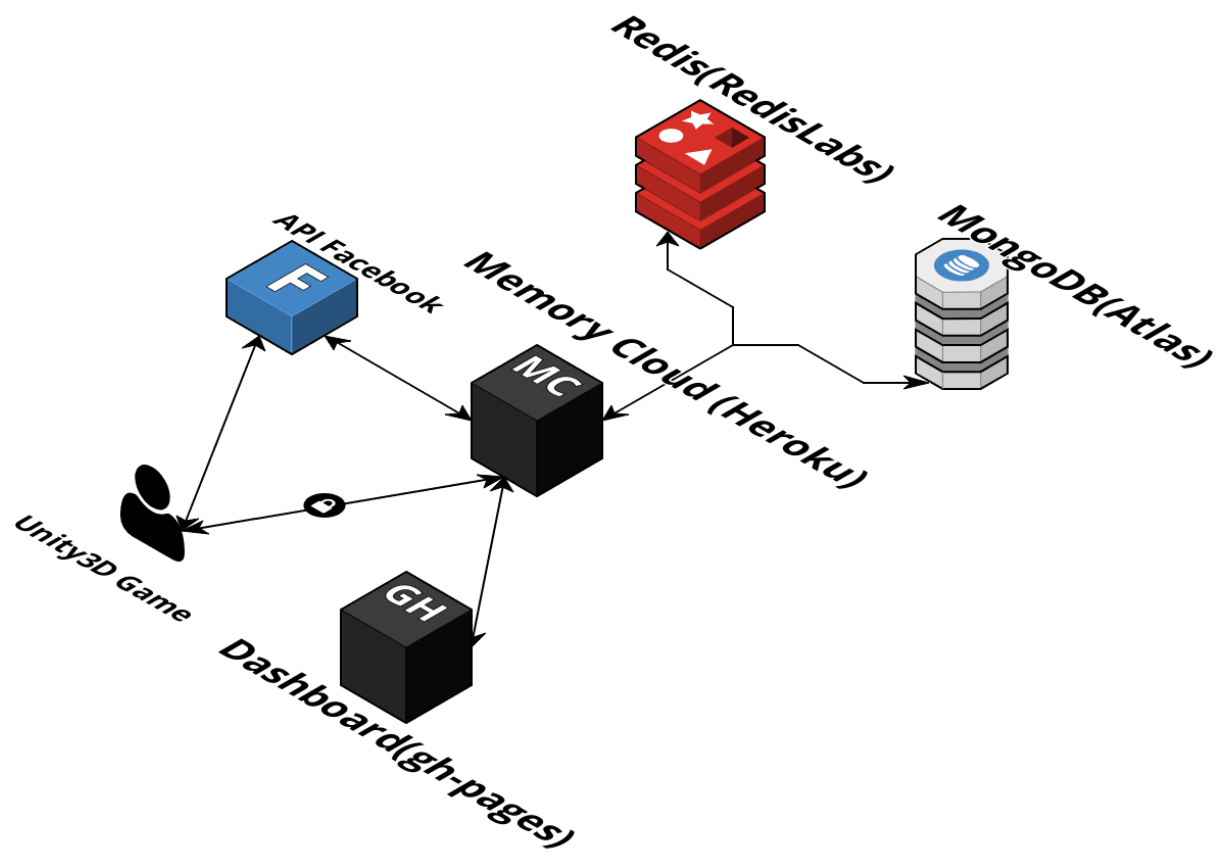
### 5.1 ARQUITETURA DO SISTEMA

#### 5.1.1 Arquitetura cliente/servidor Multicamadas



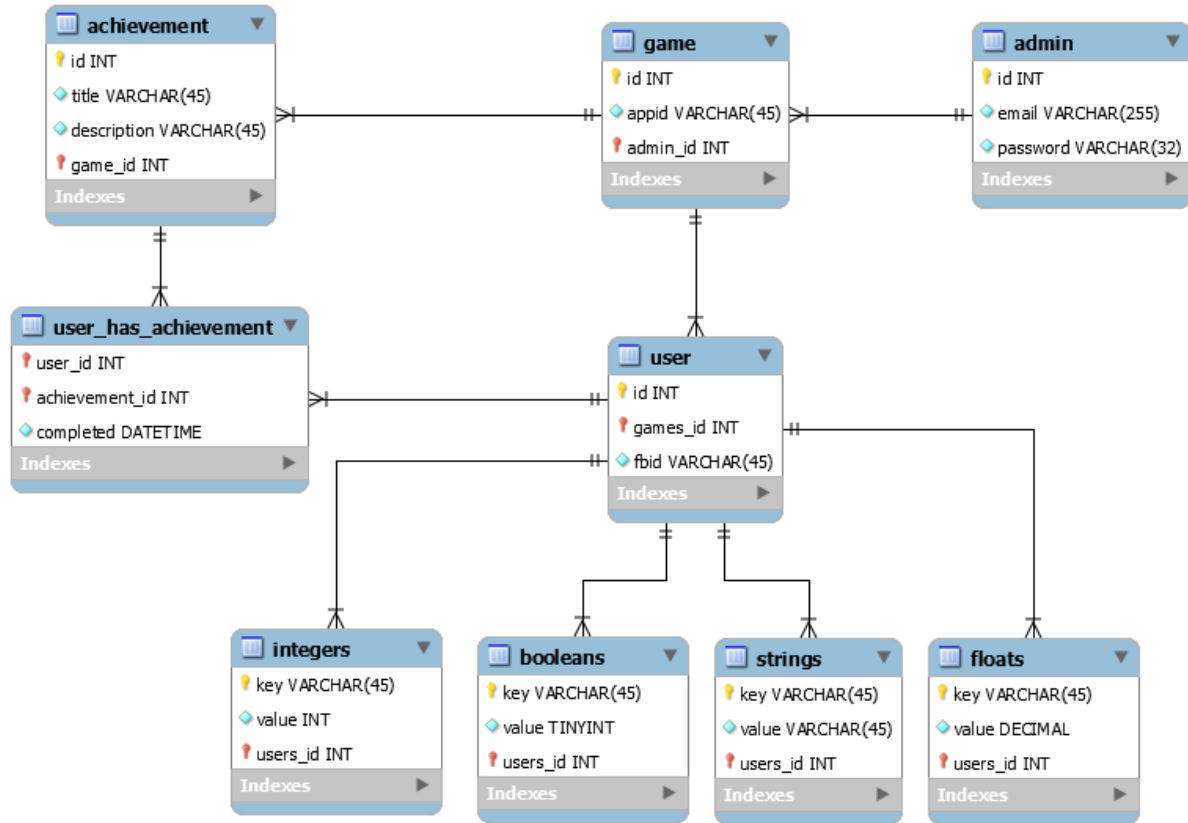
Fonte: (IBM, 2018)

## 5.1.2 Infraestructura



## 5.2 MODELO DO BANCO DE DADOS

### 5.2.1 Modelo Lógico



### 5.2.2 Dicionário de Dados

#### 5.2.2.1 Entidade: admin

Nome	Tipo	Tamanho	Descrição	Obrigatório
id	ObjectID			Sim
email	String	100		Sim
password	String	200	Hash usando bcrypt	Sim

#### 5.2.2.2 Entidade: achievement

Nome	Tipo	Tamanho	Descrição	Obrigatório
id	ObjectID			Sim
title	String	200		Sim
description	String	200		Sim
game	ObjectID			Sim



## 5.2.2.3 Entidade: game

Nome	Tipo	Tamanho	Descrição	Obrigatório
id	ObjectID			Sim
appid	String	50	Facebook App ID	Sim
key	String	200	Facebook App Token	Sim
admin	ObjectID			Sim
achievements	[ObjectID]			Sim

## 5.2.2.4 Entidade: integerDict

Nome	Tipo	Tamanho	Descrição	Obrigatório
key	String	100		Sim
value	Integer			Sim

## 5.2.2.5 Entidade: floatDict

Nome	Tipo	Tamanho	Descrição	Obrigatório
key	String	100		Sim
value	Float			Sim

## 5.2.2.6 Entidade: stringDict

Nome	Tipo	Tamanho	Descrição	Obrigatório
key	String	100		Sim
value	String	200		Sim

## 5.2.2.7 Entidade: boolDict

Nome	Tipo	Tamanho	Descrição	Obrigatório
key	String	100		Sim
value	Bool			Sim

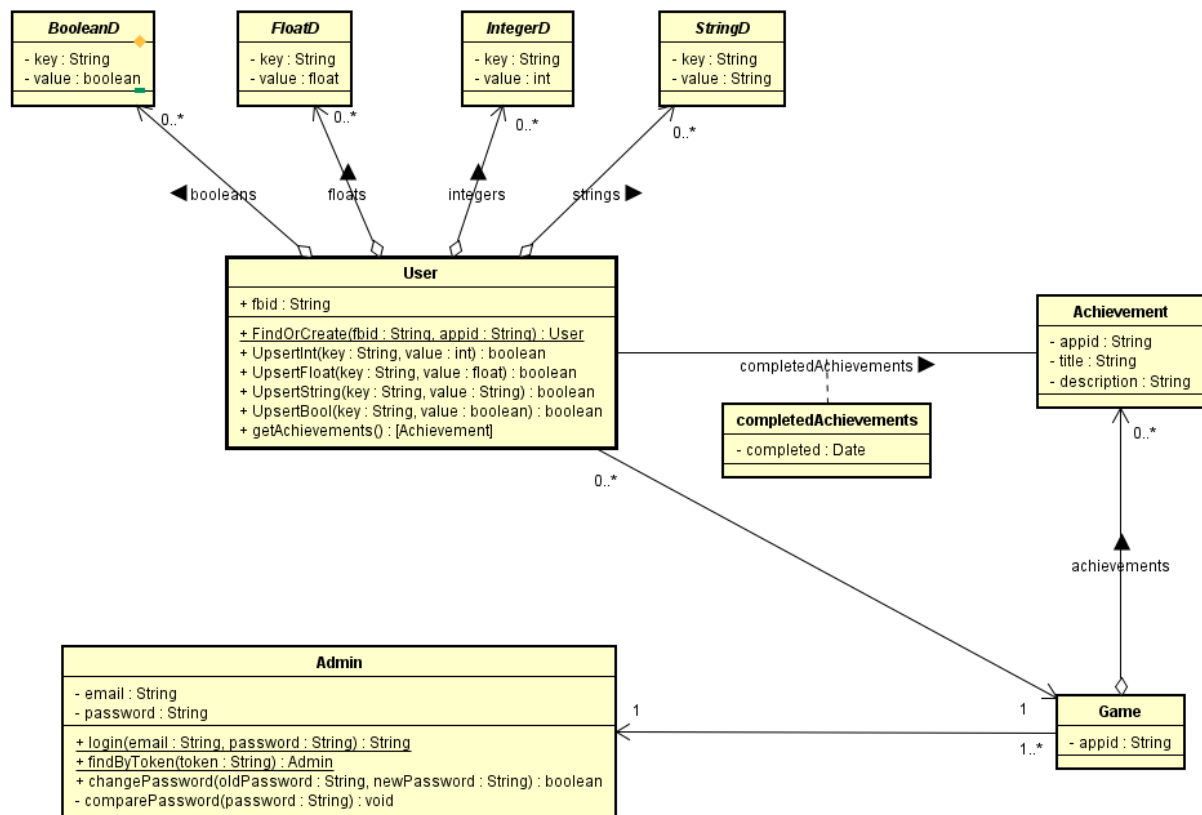
## 5.2.2.8 Entidade: achievementDone

Nome	Tipo	Tamanho	Descrição	Obrigatório
id	ObjectID		Achievement ID	Sim
date	DateTime			Sim

## 5.2.2.9 Entidade: user

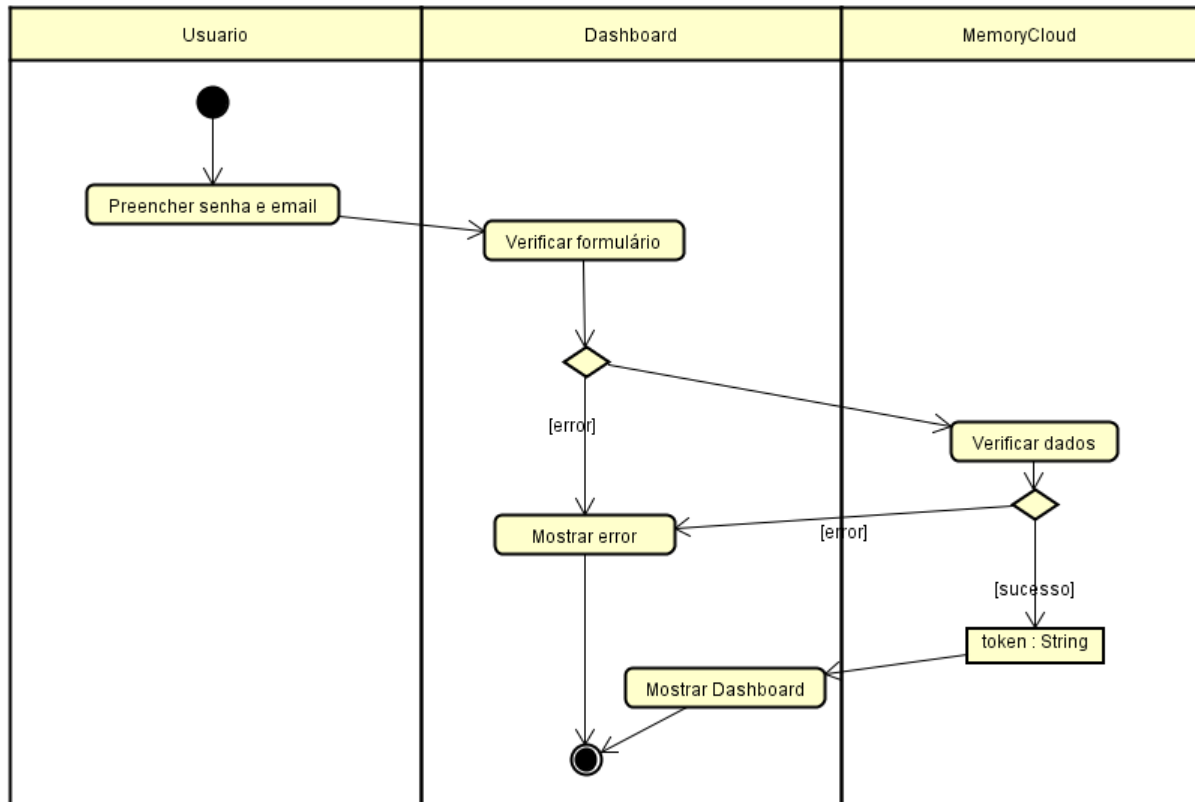
Nome	Tipo	Tamanho	Descrição	Obrigatório
id	ObjectID			Sim
fbid	String	100	Facebook Scoped ID	Sim
game	ObjectID			Sim
strings	[stringDict]		Array de stringDict	Não
integers	[integerDict]		Array de integerDict	Não
floats	[floatDict]		Array de floatDict	Não
booleans	[boolDict]		Array de boolDict	Não
achievements	[achievementDone]		Array de achievements	Sim

### 5.3 DIAGRAMA DE CLASSES

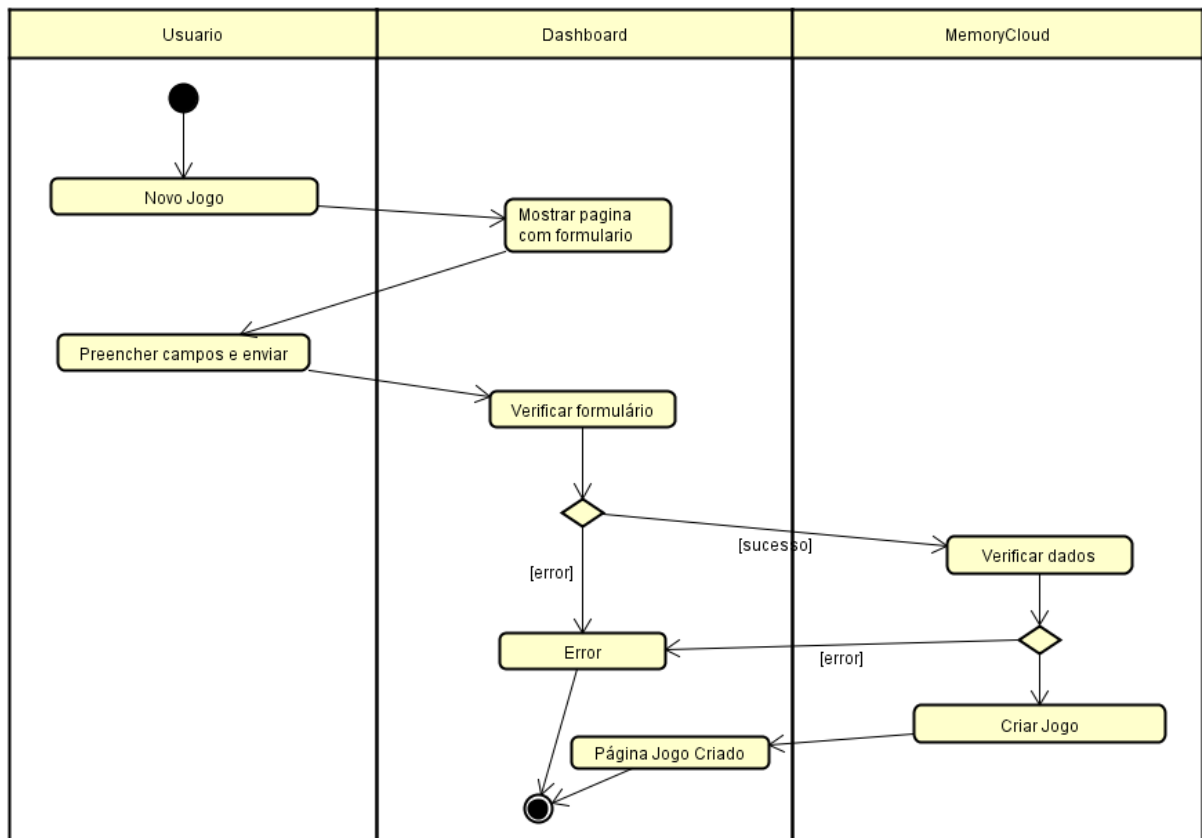


## 5.4 DIAGRAMA DE ATIVIDADES

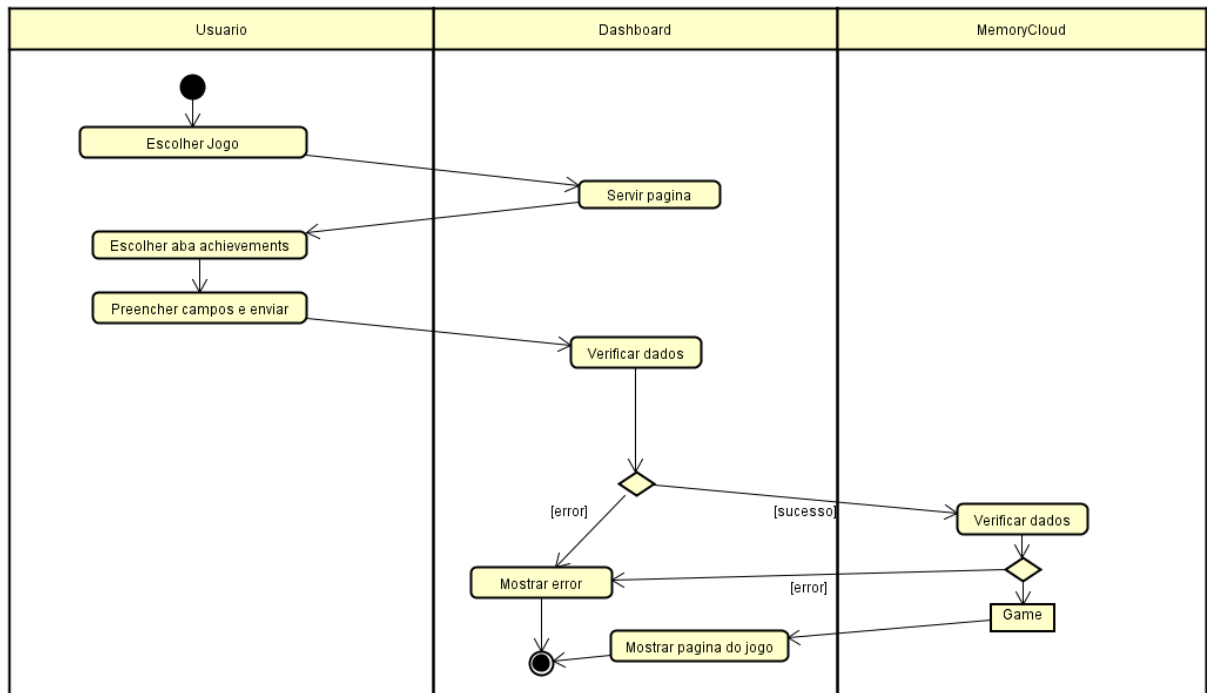
### 5.4.1 Login



## 5.4.2 Novo Jogo

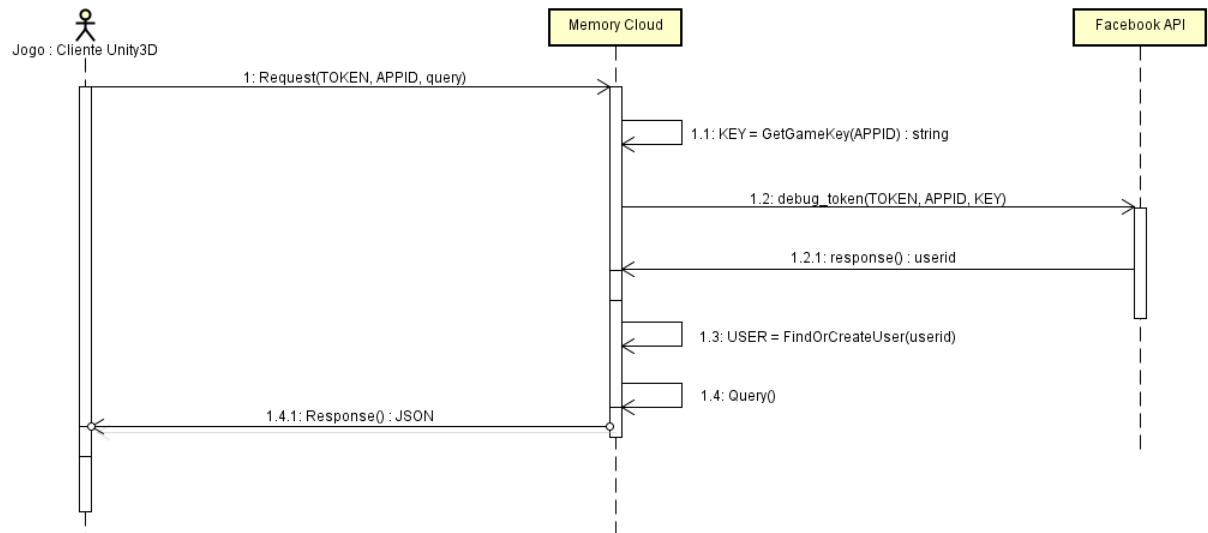


## 5.4.3 Gerenciar Achievements



## 5.5 DIAGRAMA DE SEQUENCIA

### 5.5.1 Requisição

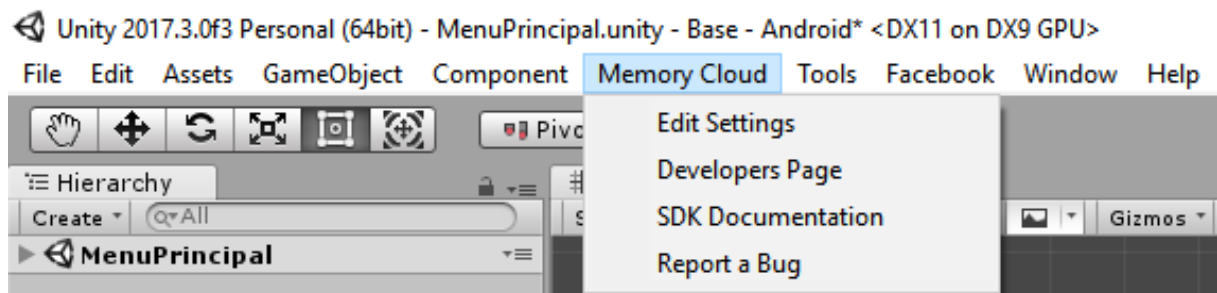


## 6 IMPLEMENTAÇÃO

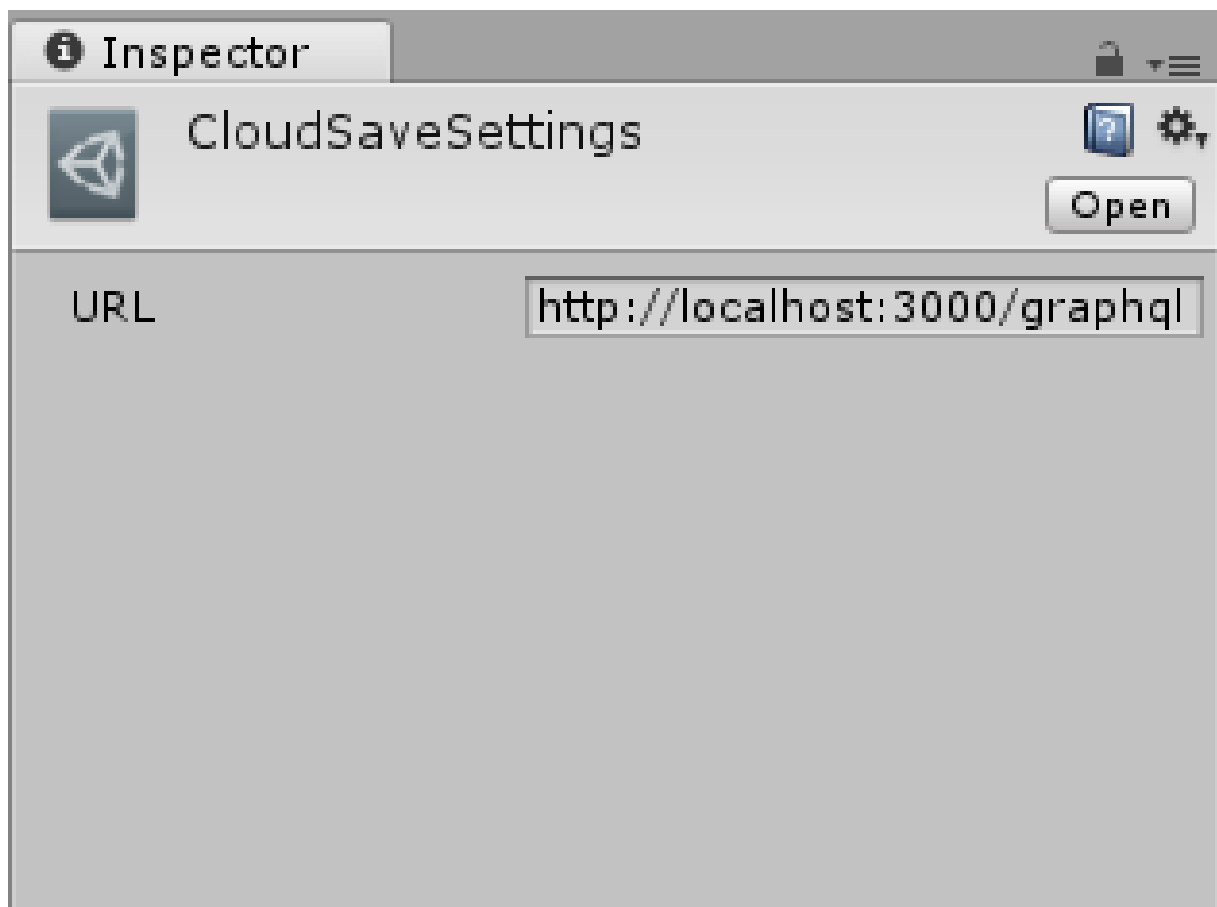
### 6.1 PROTÓTIPO DE TELAS

#### 6.1.1 Unity

##### 6.1.1.1 Menu



##### 6.1.1.2 Settings





## 6.1.2 Dashboard

### 6.1.2.1 Login

Memory Cloud

Login

[↑ need to create an account?](#)

### Login

© Memory Cloud 2018

### 6.1.2.2 SignUp

Memory Cloud

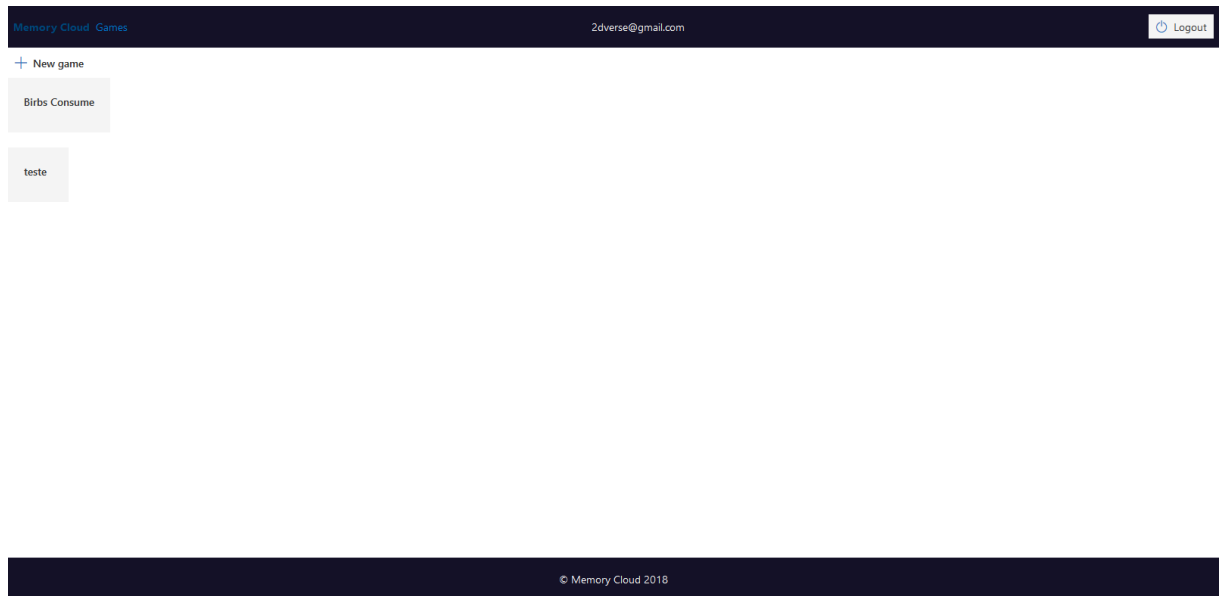
Login

[□ already have an account?](#)

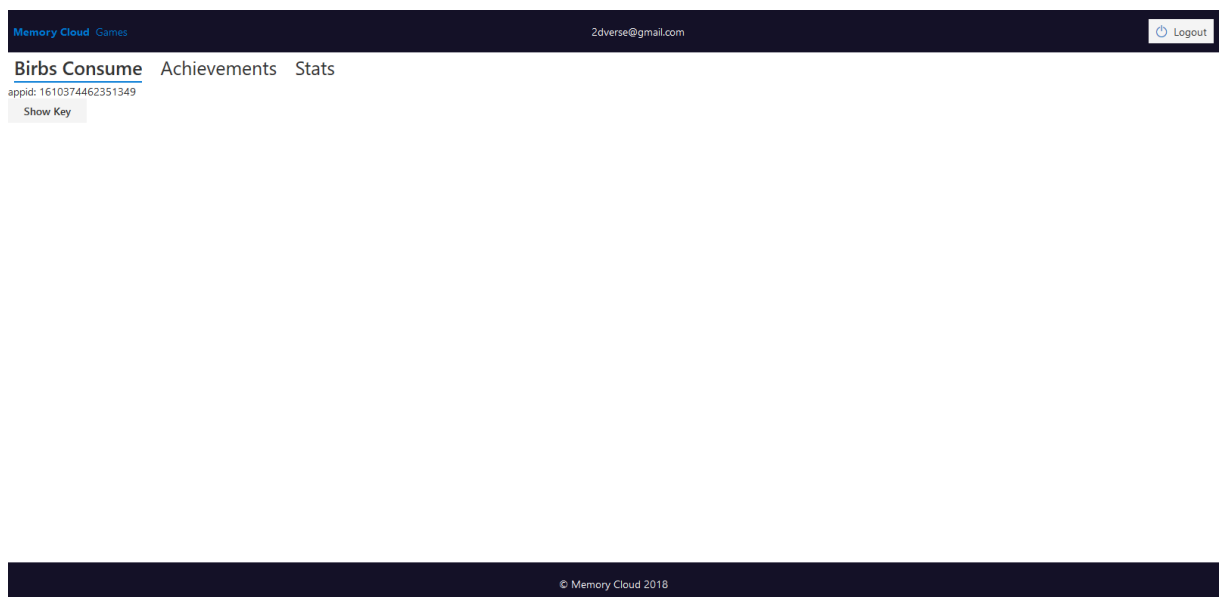
### Sign Up

© Memory Cloud 2018

### 6.1.2.3 Games



### 6.1.2.4 Game



### 6.1.2.5 Achievements

Memory Cloud Games

2dverse@gmail.com

Logout

Birbs Consume

Achievements

Stats

+ new

Ach1\_Title

Ach1\_Description

Ach2\_Title

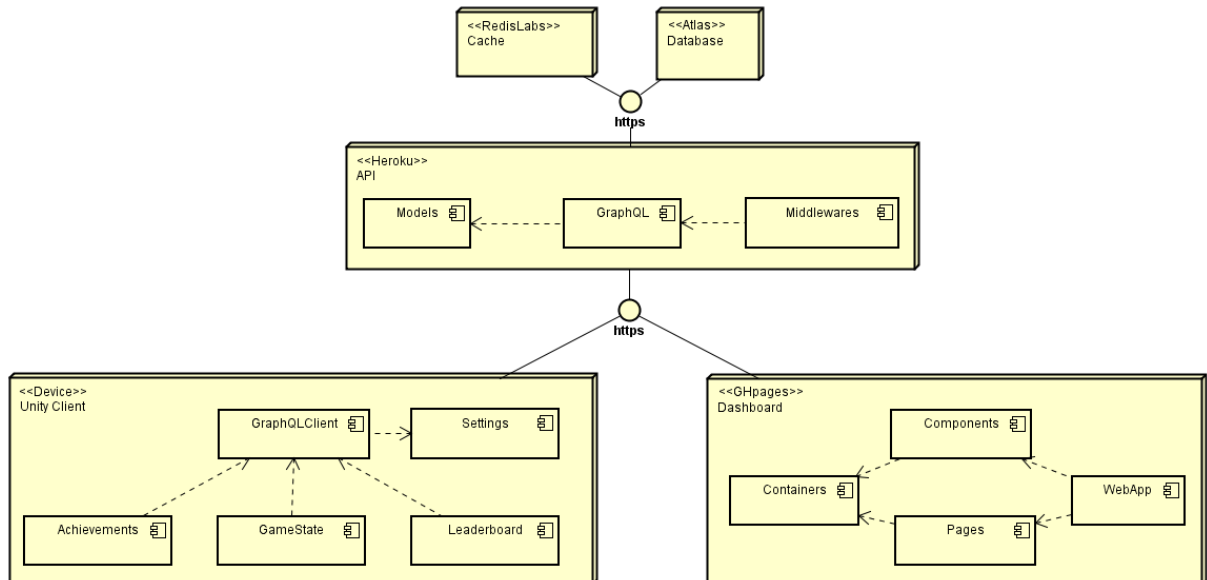
Ach2\_Description

save achievements

© Memory Cloud 2018

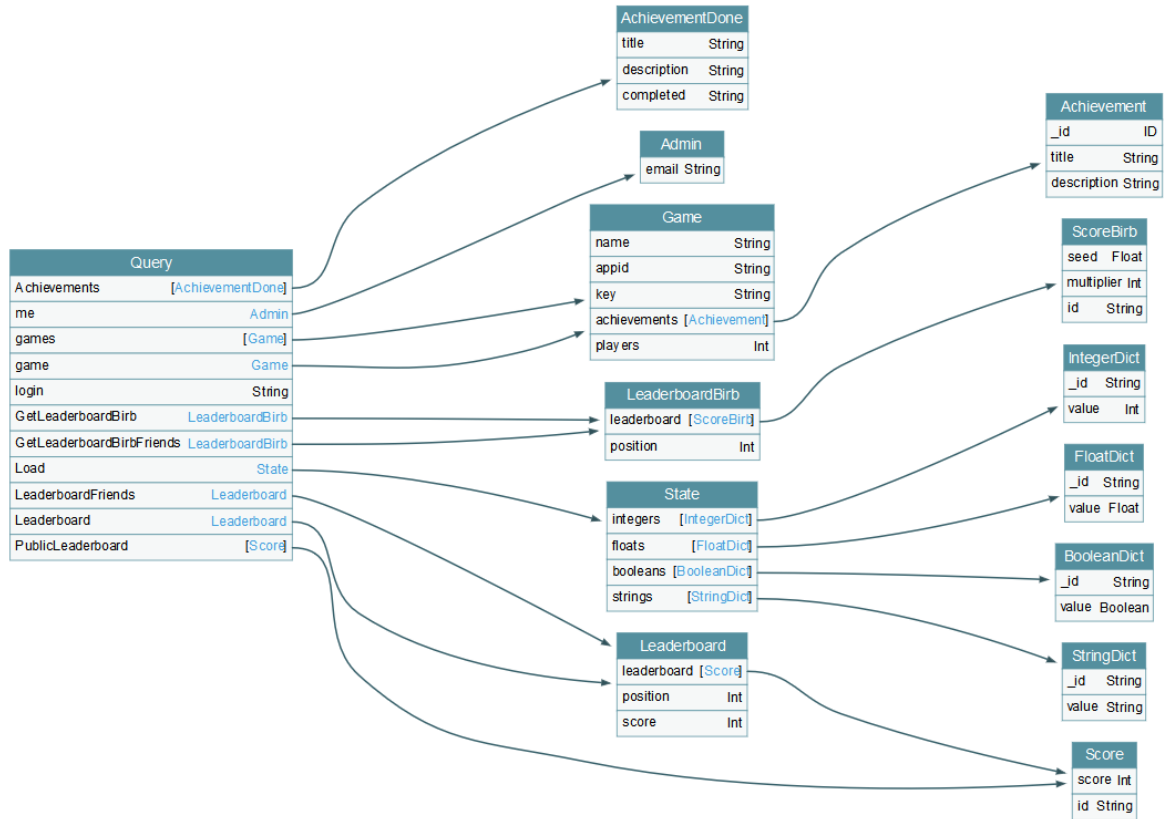
## 6.2 DESCRIÇÃO DO CÓDIGO

### 6.2.1 Diagrama de Implementação

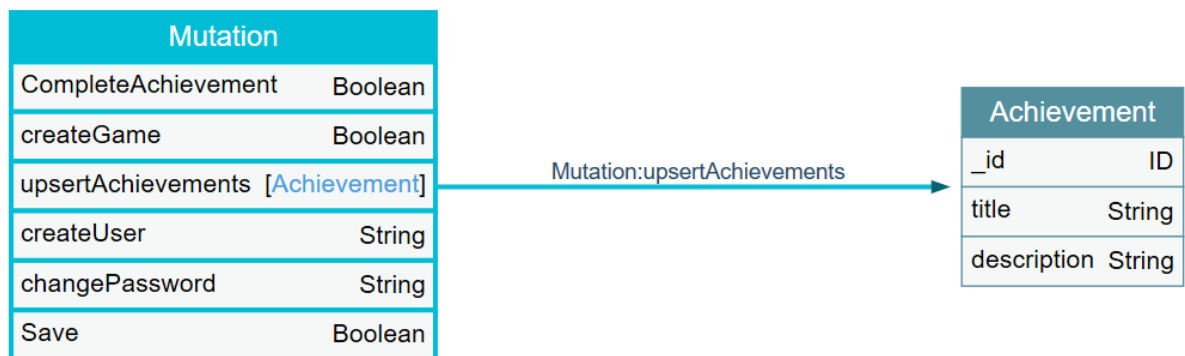


## 6.2.2 Esquema GraphQL

### 6.2.2.1 Query



### 6.2.2.2 Mutation



## **7 CONSIDERAÇÕES FINAIS**

## REFERÊNCIAS

FACEBOOK. **Breaking Changes**. 2018. Disponível em: <<https://developers.facebook.com/docs/graph-api/changelog/breaking-changes/#4-4-2018>>. Acesso em: 9 de abril de 2018. Citado na página 4.

GOOGLE. **Focusing our Google Play games services efforts**. 2017. Disponível em: <<https://android-developers.googleblog.com/2017/04/focusing-our-google-play-games-services.html>>. Acesso em: 9 de abril de 2018. Citado na página 4.

IBM. **Three-tiered client/server architecture**. 2018. Disponível em: <[https://www.ibm.com/support/knowledgecenter/en/SSAL2T\\_8.1.0/com.ibm.cics.tx.doc/concepts/c\\_three\\_tierd\\_clnt\\_sevr\\_arch.html](https://www.ibm.com/support/knowledgecenter/en/SSAL2T_8.1.0/com.ibm.cics.tx.doc/concepts/c_three_tierd_clnt_sevr_arch.html)>. Acesso em: 4 de junho de 2018. Citado na página 13.

SAGA. **O mercado que mais cresce no mundo**. 2018. Disponível em: <<https://saga.art.br/desenvolvimento-de-jogos-o-mercado-que-mais-cresce-no-mundo/>>. Acesso em: 9 de abril de 2018. Citado na página 3.

SCRUM. **Scrum Framework**. 2018. Disponível em: <<https://www.scrum.org/resources/what-is-scrum>>. Acesso em: 9 de abril de 2018. Citado na página 9.