

## **Documento de Requisitos**

**Gacha (Nome a definir)**

**Juiz de Fora, 2023**

# SUMÁRIO

<b>1. Introdução.....</b>	<b>2</b>
<b>1.1. Propósito.....</b>	<b>2</b>
<b>1.2. Escopo.....</b>	<b>3</b>
<b>1.3. Definições e abreviações.....</b>	<b>3</b>
<b>1.4. Visão geral do documento.....</b>	<b>3</b>
<b>2. Descrição geral.....</b>	<b>4</b>
<b>2.1. Perspectivas do produto.....</b>	<b>4</b>
<b>2.2. Funções do produto.....</b>	<b>4</b>
<b>2.3. Características do usuário.....</b>	<b>4</b>
<b>3. Requisitos funcionais.....</b>	<b>4</b>
<b>3.1. Front-end.....</b>	<b>4</b>
<b>3.2. Back-end.....</b>	<b>5</b>
<b>4. Requisitos não funcionais.....</b>	<b></b>

## 1. Introdução:

### 1.1. Propósito:

O propósito maior do projeto é capacitar os alunos de Engenharia de Software a desenvolver habilidades e competências necessárias para se tornarem profissionais altamente qualificados na área. Para alcançar esse objetivo, o projeto busca tornar a aprendizagem mais engajadora e eficaz, proporcionando aos alunos experiências práticas e interativas que os motivem a explorar e aprofundar seus conhecimentos.

### 1.2. Escopo:

O escopo do projeto é desenvolver um aplicativo gacha voltado para estudantes de engenharia da computação, que visa tornar a aprendizagem mais interativa e engajadora. O objetivo principal é incentivar os alunos a estudarem por meio de quizzes que concedem tiros

para serem utilizados no banner, onde podem ser obtidos personagens e itens relacionados a cada um deles.

O aplicativo conta com três personagens e três itens relacionados a cada um deles, que podem ser equipados para aumentar a quantidade de tiros concedidos ao jogar o quiz. O jogo é liberado em capítulos, que são liberados pelo professor da matéria conforme avança no conteúdo programático.

As principais funcionalidades do aplicativo são: quizzes que concedem tiros para serem utilizados no banner, banner que pode ser utilizado para obter personagens e itens relacionados à área, capítulos liberados pelo professor que avança no conteúdo programático, personagens e itens que podem ser equipados para aumentar a quantidade de tiros concedidos ao jogar o quiz. O aplicativo é projetado para ser interativo e incentivar os alunos a estudarem enquanto se divertem.

### 1.3. Definições e abreviações:

- **RF:** requisito funcional;
- **RNF:** requisito não funcional.

### 1.4. Visão geral do documento:

- **Seção 2 - Descrição Geral:** apresenta uma visão geral do sistema, especificando a perspectiva do produto e detalhamento do escopo do sistema através da discretização das funções do produto. Além disso, são explicitadas as características gerais dos usuários do produto e as restrições que poderão limitar as possibilidades de desenvolvimento.
- **Seção 3 - Requisitos funcionais:** apresentação de todos os requisitos funcionais do sistema. Descreve as principais ações do produto, considerando a aceitação e processamento das entradas e o processamento e geração das saídas.
- **Seção 4 - Requisitos não funcionais:** apresentação de todos os requisitos não funcionais do sistema. Descreve todos os aspectos qualitativos do sistema, explicitando os detalhes de facilidade de uso, manutenibilidade, confiabilidade,

desempenho, segurança, distribuição, adequação a padrões e requisitos de hardware e software.

## 2. Descrição geral:

### 2.1. Github:

[https://github.com/AngryBeanTrevota/Livia\\_Rafael\\_Joao\\_Artur/tree/master](https://github.com/AngryBeanTrevota/Livia_Rafael_Joao_Artur/tree/master)

### 2.2. Trello:

<https://trello.com/b/XI5wAqp2/gacha>

## 3. Requisitos funcionais:

### 3.1. Front-end:

Requisito 01 / RF001 - Telas Login e Registro	
1. Descrição	É necessário desenvolver uma tela inicial para o sistema, que permita aos usuários acessar as funcionalidades oferecidas pelo sistema. Essa tela deverá conter as opções de Login (Aluno e Professor) e Registro de Usuário (Aluno) A tela de Login deve permitir que o usuário insira suas credenciais de acesso, enquanto a tela de Registro deve permitir que o usuário crie uma nova conta de usuário.
2. Dados	<ul style="list-style-type: none"><li>• Matrícula (string);</li><li>• Senha (string);</li></ul>

Requisito 02 / RF002 - Tela Menu Principal	
1. Descrição	O objetivo desta funcionalidade é fornecer aos usuários (Professores/Administradores e Alunos) um menu principal que apresente as opções disponíveis no sistema. Se o usuário for um Professor/Administrador, a tela deve exibir uma lista das salas que ele possui, permitindo que ele gerencie essas salas, bem como uma opção para criar uma nova sala. Se o usuário for um Aluno, a tela deve exibir a lista

	de salas disponíveis para entrar, bem como uma opção para entrar em uma dessas salas.
--	---

### **Requisito 03 / RF003 - Tela da Sala**

<b>1. Descrição</b>	<p>O objetivo desta funcionalidade é apresentar ao usuário a sala em que ele está participando. A tela deve exibir um ícone de inventário, que mostra os personagens e itens do usuário. Também deve haver um ícone que leva o usuário para a área da história do jogo. Além disso, a tela deve apresentar um ícone para permitir que o usuário acesse a área de banners, onde ele pode adquirir novos personagens e itens.</p> <p>Para o professor que criou a sala, aparecerá um ícone que leva à lista de questionários respondidos, com o desempenho de cada aluno.</p>
---------------------	---

### **Requisito 04 / RF004 - Tela do Banner**

<b>1. Descrição</b>	Será apresentado uma tela com um Banner sendo exposta na Tela da Sala na qual haverá uma animação que será utilizada para representar um ganho ou uma conquista relacionada ao personagem ou ao item em questão.
---------------------	--

### **Requisito 05 / RF005 - Tela de Inventário**

<b>1. Descrição</b>	Será feito uma Tela de Inventário na qual será apresentado uma divisão tanto para itens quanto para personagens, sendo estes ícones separados, mostrando todos os itens e personagens que o Usuário possui e adquiriu até o momento.
---------------------	--

### **Requisito 06 / RF006 - Tela de História**

<b>1. Descrição</b>	<p>O objetivo desta funcionalidade é permitir que o usuário (Aluno) progrida na história do jogo. A tela deve exibir a história em forma de texto, apresentando um enredo linear que o usuário deve seguir para avançar no jogo. Em determinados momentos, a história terá endpoints, em que o usuário será apresentado a questionários que ele deve responder para continuar na história.</p>
---------------------	--

### **Requisito 07 / RF007 - Tela de questionário individual**

1. Descrição	Logo após um endpoint na Tela de História, será liberado uma tela que mostrará o questionário feito referente aquela parte da história, com opções de escolha e uma barra de para representar o tempo restante de resposta para, em seguida, continuar o andamento da história.
--------------	---

### **Requisito 08 / RF008 - Tela de personagem e item individual**

1. Descrição	O objetivo desta funcionalidade é permitir que o usuário (Aluno) visualize as informações detalhadas dos personagens e itens que ele possui. Ao clicar em um item ou personagem, o usuário será redirecionado para uma tela individual para aquele item ou personagem. Nessa tela, ele poderá visualizar os bônus que cada elemento oferece para o usuário.
--------------	---

### **Requisito 09 / RF009 - Tela de questionários**

1. Descrição	O objetivo desta funcionalidade é permitir que o Professor visualize todos os questionários disponíveis na sala e analise o desempenho de cada aluno em cada um deles. A tela deve exibir uma lista com todos os questionários da sala, permitindo que o Professor clique em cada um deles para visualizar as respostas dadas pelos alunos.
--------------	---

### **Requisito 10 / RF010 - Telas CRUD Professor e CRUD Aluno**

1. Descrição	O objetivo desta funcionalidade é permitir que os usuários (Professores e Alunos) possam gerenciar seus dados no sistema. Para isso, é necessário desenvolver duas telas distintas: uma para o CRUD (Create, Read, Update e Delete) de Professores/Administradores e outra para o CRUD de Alunos. Essas telas de CRUD devem permitir que o usuário crie novos perfis, visualize, edite e exclua perfis existentes.
--------------	--

## 3.2. Back-end:

### Requisito 11 / RF011 - Responder Questionário

1. Descrição	O programa deve salvar os acertos de cada usuário através da quantidade de tiros que ele conseguiu ao resolver as questões (relativo aos acertos e levando em conta os itens e personagens equipados)
2. Dados	<ul style="list-style-type: none"><li>• Matrícula (String)</li><li>• Item equipado (Item)</li><li>• Personagem equipado (Personagem)</li></ul>

### Requisito 12 / RF012 - Rodar Gacha

1. Descrição	O programa deve guardar os itens e personagens obtidos no banner no inventário do usuário, e deve alterar a quantidade de tiros na medida que eles forem gastos.
2. Dados	<ul style="list-style-type: none"><li>• Matrícula (String)</li><li>• Itens do usuário (Item)</li><li>• Personagens do usuário (Personagem)</li><li>• Quantidade de tiros (Integer)</li></ul>

### Requisito 13 / RF013 - Equipar personagens/itens

1. Descrição	O programa deve mudar o registro de qual personagem/item está equipado quando o usuário equipar ou alterar algum deles.
2. Dados	<ul style="list-style-type: none"><li>• Matrícula (String)</li><li>• Item equipado (Item)</li><li>• Personagem equipado (Personagem)</li></ul>

### Requisito 14 / RF014 - Visualizar questionários/história disponíveis

1. Descrição	O programa deve verificar quais questionários já foram liberados e quais não foram, permitindo que o usuário acesse apenas os questionários que já foram abertos pelo professor.
--------------	--

2. Dados	<ul style="list-style-type: none"> <li>• Turma do usuário (Sala)</li> </ul>
----------	---

### **Requisito 15 / RF015 - Visualizar personagens/itens disponíveis**

1. Descrição	O usuário pode acessar uma área com todos os personagens e itens disponíveis(mesmo os não obtidos), e conferir algumas informações sobre eles.
2. Dados	<ul style="list-style-type: none"> <li>• Matrícula (String)</li> <li>• Itens do usuário (Item)</li> <li>• Personagens do usuário (Personagem)</li> </ul>

### **Requisito 16 / RF016 - Poder ganhar experiência para subir o nível dos personagens**

1. Descrição	Conforme o usuário avança no jogo e completa os questionários, dependendo de sua pontuação, ele receberá uma quantidade específica de experiência com o personagem que ele está equipado.
2. Dados	<ul style="list-style-type: none"> <li>• Matrícula (String)</li> <li>• Personagem equipado (Personagem)</li> </ul>

### **Requisito 17 / RF017 - Liberar trechos de interação/história com o personagem ao subir de nível**

1. Descrição	O programa deve acessar o nível do usuário e liberar diálogos específicos para cada nível para os personagens que ele tem.
2. Dados	<ul style="list-style-type: none"> <li>• Matrícula</li> <li>• Nível</li> <li>• Personagens do usuário</li> </ul>

### **Requisito 18 / RF018 - Login de Professor e Usuário**

1. Descrição	O objetivo desta funcionalidade é permitir que o usuário (Aluno ou Professor) faça login no sistema, utilizando as credenciais cadastradas anteriormente. O sistema deve dar a opção para que o usuário informe se ele é um Aluno ou um Professor. O sistema deve verificar a validade das credenciais fornecidas pelo usuário durante o processo de login, garantindo que apenas usuários autorizados possam acessar o sistema.
--------------	--



<b>2. Dados</b>	<ul style="list-style-type: none"> <li>• Tipo de Usuário (Boolean)</li> <li>• Identificador (String)</li> <li>• Senha (String)</li> </ul>
-----------------	---

### **Requisito 19 / RF019 - Liberar questionário/história**

<b>1. Descrição</b>	Quando o professor libera o questionário/história, ele deve ser disponibilizado para todos os alunos da turma.
<b>2. Dados</b>	<ul style="list-style-type: none"> <li>• Código de acesso (String)</li> <li>• Turma</li> </ul>

### **Requisito 20 / RF020 - Ler história**

<b>1. Descrição</b>	Estando logado, quando o aluno selecionar um quiz será apresentada a história do jogo após completar a história o mesmo será direcionado ao questionário.
<b>2. Dados</b>	<ul style="list-style-type: none"> <li>• Turma (Turma)</li> <li>• Matrícula</li> <li>• Questionário (Questionário)</li> </ul>

### **Requisito 21 / RF021 - Criar turma**

<b>1. Descrição</b>	Após estar logado o Professor pode adicionar uma nova sala, para tal, o mesmo deve informar o nome da sala, após isso, ficará disponível um código para que os alunos participem da sala
<b>2. Dados</b>	<ul style="list-style-type: none"> <li>• Nome da sala (String)</li> </ul>

### **Requisito 22 / RF022 - Entrar em uma turma**

<b>1. Descrição</b>	O aluno, após estar logado, deve poder entrar em uma turma por meio de um código fornecido pelo professor e, assim, depois de informar o código o aluno deverá ser vinculado à turma do respectivo professor.
<b>2. Dados</b>	<ul style="list-style-type: none"> <li>• Código (Number)</li> </ul>

### **Requisito 23 / RF023 - Acompanhar alunos da turma**

<b>1. Descrição</b>	O usuário administrador (Professor) deve poder acompanhar o desempenho dos alunos associados à turma e, portanto, deve-se listar os mesmos e de tal forma que fique
---------------------	---

	evidenciado o nível de nome do aluno, matrícula, experiência, tiros obtidos, número de questionários realizados e itens e personagens adquiridos.
<b>2. Dados</b>	<ul style="list-style-type: none"> <li>• Nível de experiência (Number)</li> <li>• Tiros obtidos (Number)</li> <li>• Nome do aluno (String)</li> <li>• Matrícula (String)</li> <li>• Número de Questionários Realizados (Number)</li> </ul>

### Requisito 24 / RF024 - CRUD Alunos

<b>1. Descrição</b>	O objetivo desta funcionalidade é permitir que o sistema realize as operações básicas de Create, Read, Update e Delete (CRUD) dos registros dos alunos. Isso inclui adicionar novos alunos, visualizar informações dos alunos existentes, atualizar informações dos alunos existentes e excluir registros de alunos. O sistema deve garantir que apenas Professores possam realizar essas operações, de forma a garantir a segurança e a integridade dos dados.
<b>2. Dados</b>	<ul style="list-style-type: none"> <li>• Nível de experiência (Number)</li> <li>• Tiros obtidos (Number)</li> <li>• Nome do aluno (String)</li> <li>• Matrícula (String)</li> <li>• Número de Questionários Realizados (Number)</li> </ul>

### Requisito 25 / RF025 - CRUD Professores

<b>1. Descrição</b>	O objetivo desta funcionalidade é permitir que o sistema realize as operações básicas de Create, Read, Update e Delete (CRUD) dos registros dos Professores. Isso inclui adicionar novos professores, visualizar informações dos professores existentes, atualizar informações dos professores existentes e excluir registros de professores. O sistema deve garantir que apenas Professores possam realizar essas operações, de forma a garantir a segurança e a integridade dos dados.
<b>2. Dados</b>	<ul style="list-style-type: none"> <li>• Código de Acesso do Professor (String)</li> <li>• Nome (String)</li> <li>• Lista de Alunos (Aluno)</li> </ul>

## 4. Requisitos não funcionais:

### **Requisito não funcional / RNF001 - *Elementos de design dinâmico e responsivo***

1.Descrição	Os elementos do app devem ser responsivos ao clique e não estáticos.
-------------	--

### **Requisito não funcional / RNF002 - *Interface intuitiva***

1.Descrição	A navegação no app deve ser simples de entender e intuitiva para o usuário.
-------------	---

### **Requisito não funcional / RNF003 - *Personalidade interessante e bem desenvolvida dos personagens***

1.Descrição	A personalidade e história dos personagens deve ser bem construída, relatable e interessante, assim como a sua relação com o personagem principal, visando criar uma conexão emocional com o usuário e assim o levando a se dedicar mais.
-------------	---

### **Requisito não funcional / RNF004 - *Estética inspirada no Windows 95 e em computadores/programas da época***

1.Descrição	A estética deve ser inspirada no Windows 95 e em programas e tecnologias da época, mas atualizado de forma que fique esteticamente agradável.
-------------	---

### **Requisito não funcional / RNF005 - *História em um cenário de desenvolvimento de IA e computação do cérebro antes da época atual***

1.Descrição	A história deve se passar em um contexto de um desenvolvimento alternativo da tecnologia, no qual avanços na área de IA e computação do cérebro avançaram mais rápido do que na nossa realidade.
-------------	--

### **Requisito não funcional / RNF006 - *Personagens refletindo defeitos do personagem principal***

1.Descrição	Cada um dos três personagens deve ter sua personalidade como pilar principal sendo algum problema do personagem principal, de modo que ao interagir e criar relações com eles, o personagem principal acabe trabalhando esses pontos nele mesmo.
-------------	--

### **Requisito não funcional / RNF007 - A mensagem principal deve ser claramente sobre o valor do esforço acima da ideia socialmente construída de “talento”**

1.Descrição	Os eventos e os conflitos dos personagens devem todos apontar para a conclusão dessa mensagem principal, visando convencer o jogador do valor da dedicação e ferramentas apropriadas acima da ideia de ser “naturalmente capaz” ou não.
-------------	---

### **Requisito não funcional / RNF008 - Segurança dos dados do usuário**

1.Descrição	O aplicativo deve ser capaz de garantir a confidencialidade, integridade e disponibilidade das informações pessoais dos usuários, protegendo-as contra acesso não autorizado, perda, roubo ou corrupção.
-------------	--

### **Requisito não funcional / RNF009 - Desempenho**

1.Descrição	O sistema deve ser capaz de suportar um grande número de usuários simultâneos sem diminuir significativamente o desempenho e tempo de resposta das funcionalidades.
-------------	---

### **Requisito não funcional / RNF010 - Código limpo e organizado**

1.Descrição	O código e suas partes devem estar bem divididos e organizados, de forma a facilitar a leitura e alteração para todos os membros do time e possíveis novos membros.
-------------	---

## 5. Gerência do Projeto - Entrega 01

### 5.1. Membros

- Rafael de Oliveira Zimbrão - 202165124AC
- João Vitor Fernandes Ribeiro Carneiro Ramos - 202165076AC
- Livia Ribeiro Pessamilio - 202165088AC
- Artur Welerson Sott Meyer - 202065552C

### 5.2. Recursos de Gestão

- Trello
- Github
- Google Docs
- Método SCRUM (Método Ágil)

### 5.3. Linguagens que serão usadas

- Javascript
- React.Js
- Node.Js
- Firebase

## 6. Cronograma Geral

Semana	Objetivos
<b>Sprint 01 (10/04)</b>	Iniciação do Projeto; RF001;
<b>Sprint 02 (17/04)</b>	RF002; RF003; RF004;
<b>Sprint 03 (24/04)</b>	RF005; RF006; RF007;
<b>Sprint 04 (01/05)</b>	RF008; RF009; RF010;
<b>Sprint 05 (08/05)</b>	<b>Revisão e testes</b>
<b>Sprint 06 (15/05)</b>	RF018; RF024; RF025;
<b>Sprint 07 (22/05)</b>	RF021; RF022;
<b>Sprint 08 (29/05)</b>	RF011; RF012;

<b>Sprint 09 (05/06)</b>	RF020; RF023;
<b>Sprint 10 (12/06)</b>	<b>Revisão e testes</b>
<b>Sprint 11 (19/06)</b>	RF013; RF015;
<b>Sprint 12 (26/06)</b>	RF014; RF019;
<b>Sprint 13 (03/07)</b>	<b>Entrega Final</b>