

Documento de  
 Visão

Sumário

[Histórico de Revisão](#_gjdgxs)

[Introdução](#_kujmq59wa1ln)

[Escopo](#_1fj465z6xmcu)

[Oportunidade de Negócio](#_pqs111plekco)

[Glossário](#_y9ac4u10qikx)

[Principais Necessidades da Parte Interessada ou do Usuário](#_8rpt9y9vbctb)

[Visão Geral do Produto](#_892cv3y7xrqz)

[Resumo das Capacidades](#_bgc0ssssvfg1)

[Suposições e Dependências](#_w8yg07t0o9dz)

[Custo e Precificação](#_t3fv47v2i1oa)

[Recursos do Produto](#_91icga1jorg5)

[Requisitos Conscientes](#_6zqcfd9iwift)

[Requisitos Subconscientes](#_spnbqhwiaxmb)

[Requisitos de Restrição](#_l6mou1xx05yi)

[Requisitos de Qualidade](#_hewtk4j33nc)

[Requisitos Inconscientes](#_k20hvfdc2q1i)

[Restrições](#_u4fmo4m8lqlu)

[Apêndice](#_w7r38s1oqez7)

[Identificação dos Riscos](#_edsnirg8q9us)

[Diagrama de Casos de Uso](#_el9y6mqwg5xx)

[Cronograma](#_4bndwegi7pgw)

# 

# 

# **Histórico de Revisão**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor** |
| 16/08 | 0.1 | Criação do documento de visão | Lucas Fernandes |
| 18/08 | 0.2 | Adição da história dos requisitos | Karina Hagiwara |
| 22/08 | 0.3 | Alteração da história dos requisitos | Karina Hagiwara |
| 23/08 | 0.4 | Organização e Completude do Documento | Guilherme Bornia |

# 

# 

# Introdução

Este documento tem como propósito especificar o que será desenvolvido, para quem e quando será entregue, além de delimitar as funções do sistema.

As partes interessadas no software estão aqui descritas e, para elas, há um glossário com as palavras comumente usadas e essenciais para compreender o objetivo do sistema. Também é feita uma análise de mercado para verificar a demanda pelo software assim como também foi feita a listagem dos requisitos extraídos do cliente por meio de uma entrevista não-estruturada e, posteriormente, de uma entrevista estruturada.

Também são estabelecidos as restrições do projeto, os riscos existentes e um diagrama de casos de uso. Dessa forma, espera-se que as partes envolvidas compreendam o projeto do sistema “RateMyCampus”.

# Escopo

O objetivo do software “RateMyCampus” é oferecer uma visão crítica da situação dos campus das universidades brasileiras através da avaliação dos próprios alunos que possuem uma vivência em determinado(s) campus.

Será desenvolvido um sistema no qual haverão as universidades e seus campus pré-cadastrados e, para o usuário conseguir avaliá-lo(s), será necessário que o mesmo se cadastre (aluno, professor ou funcionário) no sistema informando seus dados pessoais e o(s) campus em que estuda/leciona.

Quando cadastrado, o usuário conseguirá avaliar apenas o(s) campus que foram cadastrados em sua conta. Os objetos a serem avaliados incluem salas, laboratórios, professores, bibliotecas, restaurantes universitários e os cursos oferecidos por aquela unidade. Sendo que, os professores, assim como os funcionários, apenas não poderão avaliar os professores e os cursos. Quanto aos campus dos quais não faz parte, a avaliação poderá ser apenas visualizada.

É importante ressaltar que a avaliação será feita por meio de votos e que, se o usuário desejar, poderá comentar sobre aquele campus. Todos os votos e comentários serão anônimos, cabendo a equipe desenvolvedora assegurar tal sigilo.

# Oportunidade de Negócio

RateMyCampus é destinado a todos que vivenciam seu dia a dia em uma universidade, seja aluno, seja funcionário, seja professor. Atualmente muitas universidades não estão agradando seus clientes e não há um meio tão eficaz no mercado brasileiro para que se possa mostrar, com indicadores, o quão precisam de atenção.

Com esse software, ao criar uma conta, o usuário poderá mensurar sua opinião sobre sua universidade. Pois ao classificar seu curso, bandeijão, professores e laboratórios gerará uma nota no qual é destinada uma média à sua universidade. Assim será possível analisar se uma universidade é um bom ambiente para conviver, se um curso é bem ministrado, se alguns professores possuem uma boa didática e até comparar universidades e cursos.

# Glossário

|  |  |
| --- | --- |
| **Termo** | **Descrição** |
| Milestone | É uma [técnica](https://pt.wikipedia.org/wiki/T%C3%A9cnica) de [gerência de projetos](https://pt.wikipedia.org/wiki/Ger%C3%AAncia_de_projetos) que permite o teste da [funcionalidade](https://pt.wikipedia.org/wiki/Funcionalidade) de um novo [produto](https://pt.wikipedia.org/wiki/Produto) ao longo do [projeto](https://pt.wikipedia.org/wiki/Projeto). Não é uma atividade e não possui duração. |
| Software | É uma sequência de instruções escritas para serem interpretadas por um computador com o objetivo de executar tarefas específicas. Também pode ser definido como osprogramas que comandam o funcionamento de um computador. |
| Backup | Termo em inglês que tem o significado de cópia de segurança. É frequentemente utilizado em informática para indicar a existência de cópia de um ou mais arquivos guardados em diferentes dispositivos de armazenamento. Se, por qualquer motivo, houver perda dos arquivos originais, a cópia de segurança armazenada pode ser restaurada para repor os dados perdidos. |

# Principais Necessidades da Parte Interessada ou do Usuário

As partes envolvidas no desenvolvimento do software “RateMyCampus” são clientes universitários os quais procuraram uma equipe para construir um sistema que pudesse cumprir com os seus requisitos.

Os clientes são dois estudantes: Victor Hugo e André Gonçalves - enquanto a equipe é composta por oito membros cujos nomes são Guilherme Bornia, Luis Ricardo, Ana Carolina Pedroso, Lucas Fernandes, Karina Hagiwara, Vitor Artoni, Mirelle Bueno e João Victor Ignácio.

Foram levantadas as seguintes perguntas para extrair as necessidades das partes interessadas e através delas, foi elaborada uma tabela para facilitar a leitura:

* Quais são os motivos para o problema?
* Como o problema é resolvido agora?
* Quais soluções a parte interessada deseja?

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Necessidade** | **Prioridade** | **Interesse** | **Solução Atual** | **Solução Proposta** |
|  |  |  |  |  |

# Visão Geral do Produto

## Resumo das Capacidades

## Suposições e Dependências

## Custo e Precificação

# Recursos do Produto

## Requisitos Conscientes

* + 1. A votação deve contemplar o restaurante universitário, a biblioteca, as salas de aula, os professores e o curso para fornecer indicadores de satisfação em relação a infraestrutura do local, bem como suas particularidades.
    2. A votação deve ser feita através de um questionário, ou seja, com opções pré-estabelecidas, para ser possível gerar uma média de cada opção votada, assim como gerar uma média geral.
    3. O resultado dos votos deve ser exibido por meio de símbolos (estrelas, por exemplo) para facilitar a “visualização crítica” do usuário acerca daquela unidade de ensino.
    4. A votação deve conter um campo opcional para comentários para que o usuário consiga expressar sua opinião caso o questionário não a contemple.
    5. Uma média geral das notas dadas deve estar presente no sistema para que a avaliação contemple as opiniões de todos os eleitores e, assim, mostrar um resultado justo para todos.
    6. O aluno deve poder apenas votar no(s) campus onde estuda para não influenciar nas notas de outras unidades já que o mesmo pode não conhecer outro local além do seu próprio campus. Isso traria mais precisão nos dados.
    7. Não-alunos não devem poder votar, apenas visualizar, já que desconhecem a unidade de ensino a ser votada. Com isso, seria possível gerar dados mais precisos dos campus.
    8. Cadastrar se faz necessário tanto para votar quanto para apenas visualizar para que se tenha um controle de quantos votos terão que ser processados e para se ter uma noção de quantos usuários existem no sistema.
    9. O cadastro do usuário deve exigir apenas nome, e-mail e senha daqueles que não forem alunos e exigir daqueles que são alunos, além dessas informações, o nome da faculdade em que estuda, bem como o nome do campus e o curso. Dessa forma, é possível restringir o que cada usuário tem acesso.
    10. Os votos e comentários dos usuários devem ser anônimos para que não haja medo por parte deles em votar ou expressarem suas opiniões acerca da unidade de ensino que frequentam.

## Requisitos Subconscientes

### **Requisitos de Restrição**

* + - 1. O sistema deve ser implementado para desktops (Windows 8.0 ou posterior) pois é a plataforma utilizada pelo cliente e, portanto, se faz necessário para poder obter os benefícios oferecidos pelo software.
      2. O sistema exige conexão com a Internet para que os usuários consigam se cadastrar e visualizar o seu conteúdo de qualquer lugar.
      3. Um banco de dados deve ser criado para armazenar as informações de cadastro dos usuários para que, dessa forma, seja possível identificar o perfil de cada um.
      4. Talvez seja necessário fazer uma conexão com o servidor da UNICAMP para validar as informações do cadastro de alunos por meio do RA e, assim, garantir com segurança quem são os usuários que estão votando nas unidades de ensino.
      5. O banco de dados do sistema deve ser hospedado em um servidor para que as informações dos usuários sejam armazenadas de forma mais organizada e com maior segurança.

### Requisitos de Qualidade

* + - 1. Segurança da informação dos usuários é extremamente importante para que as opiniões e comentários do sistema permaneçam anônimas.
      2. O sistema deve ter flexibilidade e desempenho para exercer suas funcionalidades com uma boa eficiência em diversos ambientes e, com isso, agradar ao usuário.

## Requisitos Inconscientes

* + 1. Um mecanismo de busca pelas unidades de ensino se faz necessário para que o usuário não encontre dificuldades em poder realizar o seu voto e/ou comentário, ou seja, para que ele encontre o que procura facilmente.

# Restrições

* 1. Deixar para desenvolver o projeto apenas as terças-feiras, durante 4 horas;
  2. Não pode dar problemas no repositório como corromper um arquivo, por exemplo;
  3. O tempo total de desenvolvimento é curto;
  4. A falta de conhecimento prático por parte dos membros pode impactar no projeto;
  5. Não pode faltar conexão à Internet;

# Apêndice

## Identificação dos Riscos

**Formulário de informações de risco**

|  |  |  |  |
| --- | --- | --- | --- |
| **Id:** 1 | **Data:** 16/08/16 | **Probabilidade:** 50% | **Impacto:** 8 (0 a 10) |
| **Descrição:** Programador se ausentar do trabalho devido à problema de saúde. | | | |
| **Mitigação:** Cuidar da saúde, não se expor a situações de risco. | | | |
| **Plano de contingência:** Escalar um programador secundário que esteja ciente do projeto. | | | |
| **Status:** Parado | | **Autor:** Mirelle Bueno | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Id:** 2 | **Data:** 16/08/16 | **Probabilidade:** 50% | **Impacto:** 9 (0 a 10) |
| **Descrição:** Servidor local apresentar defeitos ou dificuldades de acesso. | | | |
| **Mitigação:** Fazer backup diários. | | | |
| **Plano de contingência:** Utilizar servidores alternativos para acesso. | | | |
| **Status:** Parado | | **Autor:** Mirelle Bueno | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Id:** 3 | **Data:** 16/08/16 | **Probabilidade:** 60% | **Impacto:** 8 (0 a 10) |
| **Descrição:** O ambiente de trabalho ficar sem acesso a internet. | | | |
| **Mitigação:** Usar acesso de internet secundário. | | | |
| **Plano de contingência:** Utilizar os backups e adaptar o projeto. | | | |
| **Status:** Parado | | **Autor:** Mirelle Bueno | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Id:** 4 | **Data:** 16/08/16 | **Probabilidade:** 80% | **Impacto:** 08 (0 a 10) |
| **Descrição:** Computadores apresentarem defeitos | | | |
| **Mitigação:** Disponibilizar computadores alternativos para uso de desenvolvimento. | | | |
| **Plano de contingência:** Utilizar um computador alternativo. | | | |
| **Status:** Parado | | **Autor:** Mirelle Bueno | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Id:** 5 | **Data:** 16/08/16 | **Probabilidade:** 60% | **Impacto:** 04 (0 a 10) |
| **Descrição:** Problemas de comunicação com o cliente principal | | | |
| **Mitigação:** Estabelecer contatos alternativos (Ex:Skype) | | | |
| **Plano de contingência:** Entrar em contato com os usuários secundários. | | | |
| **Status:** Parado | | **Autor:** Mirelle Bueno | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Id:** 6 | **Data:** 23/08/16 | **Probabilidade:** 5% | **Impacto:** 10 (0 a 10) |
| **Descrição:** Crise econômica suspender atividades da universidade e congelar o projeto. | | | |
| **Mitigação:** Procurar por recursos externos e independentes. | | | |
| **Plano de contingência:** Analisar a situação em conjunto com o cliente | | | |
| **Status:** Parado | | **Autores:** Mirelle Bueno e Vitor Artoni | |
| **Id:** 7 | **Data:** 23/08/16 | **Probabilidade:** 50% | **Impacto:** 9 (0 a 10) |
| **Descrição:** O software não conseguir uma boa aceitação por parte da usuários. | | | |
| **Mitigação:** Requisitos levantados se aproximarem o máximo do que os usuários realmente querem. | | | |
| **Plano de contingência:** Rever o projeto e propor melhorias ao software | | | |
| **Status:** Parado | | **Autores:** Mirelle Bueno e Vitor Artoni | |

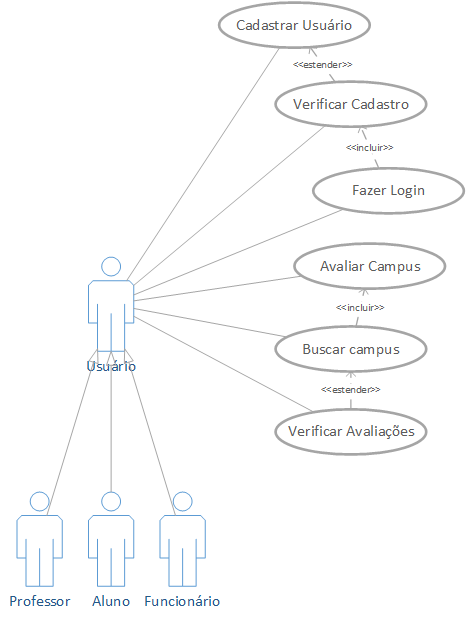
|  |  |  |  |
| --- | --- | --- | --- |
| **Id:** 8 | **Data:** 23/08/16 | **Probabilidade:** 40% | **Impacto:** 9 (0 a 10) |
| **Descrição:** Os dados anônimos das avaliações vazarem | | | |
| **Mitigação:** Elaborar mecanismo de segurança dos dados | | | |
| **Plano de contingência:** Encontrar o responsável e reforçar a segurança | | | |
| **Status:** Parado | | **Autores:** Mirelle Bueno e Vitor Artoni | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Id:** 9 | **Data:** 23/08/16 | **Probabilidade:** 70% | **Impacto:** 8 (0 a 10) |
| **Descrição:** Erro no planejamento do projeto causa atraso na entrega | | | |
| **Mitigação:** Revisar constantemente o planejamento | | | |
| **Plano de contingência:** Adaptar o máximo possível o planejamento e encurtar os prazos | | | |
| **Status:** Parado | | **Autores:** Mirelle Bueno e Vitor Artoni | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Id:** 10 | **Data:** 23/08/16 | **Probabilidade:** 30% | **Impacto:** 7 (0 a 10) |
| **Descrição:** Incompatibilidades do software com o sistema operacional | | | |
| **Mitigação:** Se atentar a questões de compatibilidade e atualizações do sistema operacional | | | |
| **Plano de contingência:** Convocar os desenvolvedores e resolver o problema | | | |
| **Status:** Parado | | **Autores:** Mirelle Bueno e Vitor Artoni | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Id:** 11 | **Data:** 23/08/16 | **Probabilidade:** 35% | **Impacto:** 5 (0 a 10) |
| **Descrição:** Risco não ser encontrado nos testes e causar transtornos ao usuário | | | |
| **Mitigação:** Certificar-se de que os testes estão cobrindo a maioria dos erros | | | |
| **Plano de contingência:** Resolver o problema o mais rápido possível | | | |
| **Status:** Parado | | **Autores:** Mirelle Bueno e Vitor Artoni | |

## Diagrama de Casos de Uso



## Cronograma

|  |  |  |
| --- | --- | --- |
| **Semana** | **Data** | **Tema** |
| 4 | 23/08 | Milestone (M1): Definição dos objetivos do ciclo de vida |
| 8 | 20/09 | Milestone (M2): Ciclo de vida da arquitetura |
| 13 | 25/10 | Milestone (M3): Capacidade operacional inicial |
| 14 | 01/11 | Milestone (M4): Release do produto |