

Projeto e Arquitetura de Software

Mathias Gatti, Rodrigo Rosa, Carlos Sousa e Ramon Correa

Graduação em Engenharia de Software

Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS) – Porto Alegre, RS – Brazil

{mathias.elbern, rodrigo.rosa.005, carlos.sousa, ramon.correa}@acad.pucrs.br

Resumo. O objetivo desse trabalho é propor o uso de um dos estilos arquiteturais estudados até agora para gerenciar inscrições feitas, validar os times criados e gerenciar as fichas de avaliação dos resultados, além de gerar informações para o certificado de participação da Hackatona do curso de Engenharia de Software no segundo semestre de 2019. A partir dos requisitos funcionais apresentados pela professora, o grupo deve produzir algo que contemple os mesmos da melhor maneira possível. O atributo de qualidade escolhido como foco do time no desenvolvimento do trabalho foi a usabilidade.

Documento de Requisitos

HACKATON MANAGEMENT SYSTEM

Histórico de revisões

Versões	
Versão 1.0	

Equipe Responsável pela Elaboração

Mathias Gatti Elbern
Ramon Correa
Carlos Sousa
Rodrigo Rosa

Público Alvo

Este sistema destina-se aos alunos interessados na Hackatona da Engenharia de Software e aos professores organizadores. O manual abaixo é voltado para pessoas interessadas em entender o projeto, tanto no âmbito de negócios como no desenvolvimento do código.

Sumário

Introdução..... P2

Visão geral deste documento..... P2

CAPÍTULO 1 - Descrição geral do sistema..... .C1 . P2

CAPÍTULO 2 - Requisitos funcionais (casos de uso)..... C2 . P2

CAPÍTULO 3 - Histórias de Usuário..... C2 . P2

CAPÍTULO 4 - Requisitos não funcionais..... C3 . P2

Introdução

Este documento tem o objetivo de especificar o sistema **Hackaton Management System**, fornecendo aos desenvolvedores as informações necessárias para execução do projeto e implementação, assim como para realizar os testes e validações do sistema descrevendo como o sistema deve funcionar. Além disso, o documento visa apresentar para qualquer pessoa interessada em continuar o projeto, uma visão completa do sistema, desde os seus requisitos não-funcionais até uma descrição da interface de usuário. O foco do grupo como atributo de qualidade escolhido foi a usabilidade do sistema, de modo que o sistema se torne o mais fácil e adaptável possível para os usuários, diminuindo a curva de aprendizado e aumentando a adoção do sistema.

1.Descrição geral do sistema

Este é um software dependente de um computador com acesso a internet para funcionamento. Por meio de uma conexão de rede, os clientes se comunicam com o servidor e acessam o sistema com a matrícula, após se cadastrarem, colocando também um nome, curso e se é usuário do tipo aluno. Cada usuário cadastrado deve ter sua matrícula cadastrada no banco de consulta da aplicação(Banco da PUCRS). Os dois tipos de papéis são os alunos e os avaliadores. As funções principais do software do ponto de vista do aluno são criar, editar, excluir ou entrar em um time selecionado entre os alunos já cadastrados na plataforma. A função principal dos avaliadores é gerenciar as fichas de avaliação dos times, votando em cada time de acordo com a apresentação do trabalho.

2.1 Resumo dos usuários:

- **Alunos:** Alunos da pucrs interessados em participar da hackatona de engenharia de software.
- **Avaliadores:** Professores e outras pessoas que serão responsáveis por avaliar os grupos na hackatona.

2.1 Ambiente de operação

O software do sistema será livre, portanto gratuito.

2.1 Pressupostos e dependências

2. Requisitos funcionais (casos de uso)

ID	Descrição
[RF-001]	Cadastrar usuários, através de conexão com WebSite de eventos como eventbrite, ou por cadastro(Matricula, nome e curso) indicando se é aluno ou não.
[RF-002]	Os usuários devem fazer o login utilizando sua matrícula.
[RF-003]	Ao entrar no sistema aluno deve poder visualizar os times e os alunos.
[RF-004]	Aluno pode criar e excluir um time, bem como incluir e excluir integrantes.
[RF-005]	Sistema deve informar ao aluno se o time é válido ou não.
[RF-006]	Ao entrar no sistema o Avaliador deve poder visualizar os times e os alunos inseridos em cada time.
[RF-006]	Avaliadores podem fazer a avaliação dos times, considerando a apresentação dos mesmos.
[RF-007]	Avaliadores devem poder avaliar os alunos nos seguintes critérios: software funcionando, processo, pitch, inovação e formação do time.
[RF-008]	Os avaliadores podem voltar de 0 - ruim a 5 - excelente para cada critério.

3. User Stories

User Story: [US-001]

Referência: [RF-002]

EU COMO usuário

QUERO fazer o login com minha matrícula

PARA acessar o sistema

Cenário 01: Matrícula existe no BD

DADO que o usuário está na tela de cadastro

QUANDO usuário digita matrícula que existe no banco de dados

ENTÃO o sistema deve abrir na página inicial de acordo com a categoria de usuário(aluno ou avaliador)

Cenário 02: Matrícula não existe no BD

DADO que o usuário está na tela de cadastro

QUANDO usuário digita matrícula não existe no banco de dados

ENTÃO o sistema deve mostrar uma mensagem de erro impedindo o usuário de logar

User Story: [US-002]

Referência: [RF-003]

EU COMO usuário do tipo aluno

QUERO visualizar os times e os alunos cadastrados

PARA decidir se quero entrar em algum time ou criar um novo.

Cenário 01: Há diversos alunos cadastrados e alguns times

DADO que o usuário aluno fez o login no sistema

QUANDO abrir a página inicial do tipo aluno

ENTÃO o aluno pode visualizar os alunos já cadastrados e os times criados por eles.

Cenário 02: Não há nenhum aluno cadastrado

DADO que o usuário aluno fez o login no sistema

QUANDO abrir a página inicial do tipo aluno

ENTÃO o sistema deve mostrar uma mensagem pedindo que o aluno crie um time inicial permitindo que ele adicione apenas ele.

User Story: [US-003]

Referência: [RF-004]

EU COMO usuário do tipo aluno

QUERO visualizar o curso de cada aluno cadastrado

PARA saber quais os cursos faltam em cada time ou quais alunos posso escolher para o meu time

Cenário 01: Aluno visualiza os outros alunos disponíveis para escolha do seu time

DADO que o aluno entrou na página inicial

QUANDO ele for selecionar os alunos para entrar no seu time

ENTÃO o sistema deve mostrar o curso de cada aluno

Cenário 02: Aluno visualiza os integrantes de um time para então entrar em um time

DADO que o aluno entrou na página inicial

QUANDO ele for selecionar um time para entrar

ENTÃO o sistema deve mostrar o curso de cada aluno inserido no time

User Story: [US-004]

Referência: [RF-004] e [RF-005]

EU COMO usuário do tipo aluno

QUERO criar um novo time

PARA participar da hackatona.

Cenário 01: Aluno escolhe 4 alunos já cadastrados de cada curso para um novo time

DADO que o aluno criou um novo time e selecionou 4 colegas de cada curso

QUANDO ele pressionar o botão salvar

ENTÃO o sistema deve mostrar uma mensagem de time criado com sucesso

Cenário 02: Aluno escolhe todos os alunos do mesmo curso para o time

DADO que o aluno criou um novo time e selecionou alguns colegas, todos do mesmo curso

QUANDO ele clicar no botão salvar

ENTÃO o sistema deve mostrar uma mensagem dizendo “Infelizmente esse time não é válido, selecione no mínimo alunos de 2 cursos diferentes para o time”.

User Story: [US-005]

Referência: [RF-006]

EU COMO usuário do tipo avaliador

QUERO visualizar os times e seus componentes

PARA gerenciar a hackatona

Cenário 01: Avaliador entra na página inicial do sistema e visualiza os times cadastrados

DADO que o Avaliador logou com sucesso no sistema

QUANDO ele entrar na página inicial

ENTÃO o sistema deve mostrar os times e os alunos inseridos em cada time

Cenário 02: Avaliador entra na página inicial do sistema e não há alunos cadastrados

DADO que o Avaliador logou com sucesso no sistema

QUANDO ele entrar na página inicial

ENTÃO o sistema deve mostrar uma mensagem que não há alunos cadastrados até o momento na hackatona.

User Story: [US-006]

Referência: [RF-007]

EU COMO usuário do tipo avaliador

QUERO avaliar os times

PARA determinar um campeão da hackatona

Cenário 01: Avaliador entra na área de votação do time

DADO que o Avaliador logou com sucesso no sistema

QUANDO ele entrar na área de votação dos times

ENTÃO o sistema deve mostrar os times e o avaliador deve ter a possibilidade de escolher um time para votação

Cenário 02: Avaliador entra na página inicial do sistema e não há times cadastrados

DADO que o Avaliador logou com sucesso no sistema

QUANDO ele entrar na página de votação

ENTÃO o sistema deve mostrar uma mensagem que não há times cadastrados até o momento na hackatona.

User Story: [US-007]

Referência: [RF-008]

EU COMO usuário do tipo avaliador

QUERO escolher um time para avaliar

E visualizar os 5 atributos de qualidade

PARA votar em cada time de acordo com a apresentação

Cenário 01: Avaliador entra na área de votação do time

DADO que o Avaliador selecionou um time para votar

QUANDO entrar na área de votação de cada time

ENTÃO o sistema deve mostrar os 5 atributos de qualidade que deve ser votados pelo avaliador.

User Story: [US-008]

Referência: [RF-009]

EU COMO usuário do tipo avaliador

QUERO escolher um dos atributos de qualidade para votar

E escolher entre as notas 0 e 5

PARA avaliar a apresentação dos grupos

Cenário 01: Avaliador entra no atributo de qualidade para votar na apresentação

DADO que o Avaliador selecionou um atributo de qualidade para votar

QUANDO ele selecionar o atributo para votar

ENTÃO o usuário deve poder selecionar uma nota de 0 a 5 para cada atributo.

4.Requisitos não funcionais

ID	Atributo de qualidade	Descrição
[RNF-001]	Portabilidade	O sistema deve funcionar em qualquer browser conectado na internet.
[RNF-002]	Confiabilidade	Toda vez que um usuário gerenciar de qualquer maneira um time(incluir, excluir usuários) o sistema deve atualizar os times para verificar se houve adições ou deleções simultâneas
[RNF-003]	Usabilidade	O sistema deve ser fácil do usuário aprender, sem levar mais de 5 minutos para o usuário conseguir entender o sistema e fazer as ações propostas a ele.
[RNF-004]	Manutenabilidade	Um programador com o mínimo de experiência em Arquitetura de Software, Springboot e Node, deve ser capaz de dar suporte ao sistema.
[RNF-005]	Integridade	Apenas usuários do tipo Avaliadores podem avaliar os times da hackatona.
[RNF-006]	Integridade	Apenas usuários do tipo Aluno podem criar e se incluir em times.
[RNF-007]	Segurança	Para logar no sistema usuário deve inserir corretamente matrícula.
[RNF-008]	Reusabilidade	O sistema deve ser desenvolvido da maneira mais

		desacoplada possível para permitir melhor reutilização das classes.
[RNF-009]	Usabilidade	Interface mais padronizada possível para agilizar a utilização do sistema pelo usuário.
[RNF-010]	Usabilidade	Custos reduzidos de apoio ao usuário, tais como treinamento, ou atendimento ao usuário, devido a facilidade do uso.

Diagrama de Componentes

