

# Universidade Federal de Ouro Preto



## Engenharia de Software II

Sistema de *Evento Esportivo*

Grupo: *PCC*

Alunos:	Carlos Magalhães Silva Diego Matos Emanuel Jesus Xavier Gabriel Bicalho Ferreira Jhonatan Gomes de Souza Lucas Andrade Freitas William Gomes
Professor:	Msc prof. Johnatan Oliveira

Horário: Seg & Qua - 08:20 -10:00

Ouro Preto, 06 de Março de 2021

# Conteúdo

<b>1</b>	<b>Histórico de Revisões</b>	<b>1</b>
<b>2</b>	<b>Processo e Software</b>	<b>1</b>
<b>3</b>	<b>Cronograma</b>	<b>2</b>
<b>4</b>	<b>Levantamento de Requisitos</b>	<b>2</b>
<b>5</b>	<b>Especificação de Requisitos</b>	<b>2</b>
5.1	Requisitos Funcionais . . . . .	2
5.2	Requisitos Não Funcionais . . . . .	5
<b>6</b>	<b>Serviços</b>	<b>6</b>
6.1	Serviço de usuário . . . . .	6
6.2	Serviço de gerenciamento de eventos . . . . .	6
6.3	Serviço de Administrador . . . . .	6
<b>7</b>	<b>Plano de VVT</b>	<b>7</b>
7.1	Requisitos a serem testados . . . . .	7
7.2	Estratégias e ferramentas de teste . . . . .	8
7.3	Equipe e infra-estrutura . . . . .	8
7.4	Execução do Plano de Teste . . . . .	8
<b>8</b>	<b>Medição e Qualidade de Software</b>	<b>10</b>
<b>9</b>	<b>Observações</b>	<b>11</b>
<b>10</b>	<b>Referências</b>	<b>12</b>

# 1 Histórico de Revisões

Data	Versão	Descrição	Autor
12/03/2021	1.0	Levantamento de requisitos	Diego
16/03/2021	2.0	Descrição dos requisitos funcionais do cliente	Emanuel
18/03/2021	3.0	Definição dos requisitos não funcionais	Gabriel
19/03/2021	4.0	Definição de serviços	PCC
20/03/2021	5.0	Diagrama de casos de uso	William

Tabela 1: Revisões do Documento

## 2 Processo e Software

*”Scrum é uma estrutura leve que ajuda pessoas, equipes e organizações a gerar valor por meio de soluções adaptativas para problemas complexos. Em suma, Scrum requer um Scrum Master para promover um ambiente onde: Um Product Owner ordena o trabalho de um problema complexo em um Product Backlog. O Time Scrum transforma uma seleção do trabalho em um incremento de valor durante uma Sprint. O Time Scrum e seus stakeholders inspecionam os resultados e se ajustam para o próximo Sprint. Scrum é simples. Experimente como está e determine se sua filosofia, teoria e estrutura ajudam a atingir objetivos e criar valor. O framework Scrum é propositalmente incompleto, apenas definindo as partes necessárias para implementar a teoria Scrum. O Scrum é baseado na inteligência coletiva das pessoas que o usam. Em vez de fornecer às pessoas instruções detalhadas, as regras do Scrum orientam seus relacionamentos e interações. Vários processos, técnicas e métodos podem ser empregados na estrutura. Scrum envolve práticas existentes ou as torna desnecessárias. O Scrum torna visível a eficácia relativa do gerenciamento, ambiente e técnicas de trabalho atuais, para que melhorias possam ser feitas.”*[1]

### 3 Cronograma

Nome	Tarefa	Prazo
Fulano	Testar ferramenta proposta no plano de teste	01/01 a 10/01
Fulano2	Implementar os casos de testes e usar as ferramentas	10/01 a 18/01

Tabela 2: Cronograma

### 4 Levantamento de Requisitos

*Casos de uso:*

1. *GERENCIAR evento esportivo: Eu como usuário dono de um evento. Gostaria de GERENCIAR meu evento esportivo. Para vender ingressos do evento.*
2. *GERENCIAR usuário dono de evento esportivo: Eu como dono de evento esportivo. Gostaria de me GERENCIAR como usuário. Para utilizar as ferramentas de usuário dono de evento esportivo.*
3. *Comprar um ingresso para evento esportivo: Eu como usuário. Gostaria de visualizar eventos cadastrados. Para comprar um ingresso para o evento.*
4. *Devolver um ingresso de um evento esportivo: Eu como usuário. Gostaria de devolver um ingresso comprado de um evento cadastrados. Para receber meu dinheiro novamente*

### 5 Especificação de Requisitos

#### 5.1 Requisitos Funcionais

**RF01**– Criar conta. **Informações:** O usuário vai precisar de adicionar o seu email, CPF, data de nascimento, endereço e cadastrar uma senha. **Regras:** O sistema deve permitir que o usuário faça o cadastro.

- O sistema deve ser fazer a validação dos dados.

**RF02**– Atualizar perfil. **Informações:** O usuário poderá alterar algumas informações como o email, senha e endereço. **Regras:** O sistema deve permitir o usuário a realizar algumas atualizações.

**RF03**– Comprar Ingresso. **Informações:** O usuário poderá realizar compras de ingressos. **Regras**– A compra só poderá ser efetuada se existirem ingressos disponíveis.

- O sistema deve verificar se o pagamento foi efetuado.

**RF04**– Reembolsar ingresso. **Informações:** Em caso de cancelamento de uma compra, o usuário deverá receber o estorno do dinheiro. **Regras:** Só será reembolsado o usuário que cancelar previamente a sua compra.

**RF05**– Retornar eventos disponíveis. **Informações:** O usuário poderá pesquisar os eventos disponíveis para compra. **Regras:** A pesquisa poderá ter os seguintes filtros:

- Tema: Apresenta todos eventos esportivos relacionados ao tema escolhido;
- Distância: Apresenta todos eventos esportivos disponíveis em um determinado raio de distância;
- Valor: Apresenta todos eventos esportivos disponíveis com o preço do ingresso até um determinado valor;
- Data: Apresenta todos eventos esportivos disponíveis em uma determinada data.

**RF06**– Deletar conta. **Informações:** O usuário poderá realizar a exclusão de sua conta.

**RF07**– Redefinir Senha. **Informações:** Caso o usuário esqueça a sua senha, ele poderá redefini-la.

**RF08**– Ver histórico de eventos. **Informações:** O sistema deve permitir que o usuário veja o histórico de compras de ingressos esportivos.

**RF09**– Criar evento. **Informações:** O usuário organizador poderá criar um novo evento esportivo. **Regras:** Existe algumas regras para a criação dos eventos, são elas:

- O evento precisa ser criado com no mínimo uma semana de antecedência;
- O local do evento deve estar disponível.

**RF10**– Cancelar evento. **Informações:** O usuário divulgador pode cancelar um determinado evento. **Regras:** O sistema deverá reembolsar automaticamente os usuários que compraram ingressos para este evento.

**RF11**– Atualizar informações do evento. Caso for necessário o Usuário organizador, poderá alterar informações sobre o evento. **Regras:** As informações possíveis de serem alteradas são:

- *Local do evento;*
- *Data do evento;*
- *Horário do evento.*

**RF12**– *Ver eventos criados.* **Informações:** *O usuário organizador poderá visualizar todos eventos criados no sistema.*

**RF13**– *Receber notificação de reports.* **Informações:** *O sistema deverá encaminhar notificações e reports para o usuário administrador.* **Regras:** *As seguintes notificações deverão ser encaminhadas para o administrador:*

- *Exclusão de evento;*
- *Report sobre falhas no sistema;*
- *Report sobre estorno não efetuado;*
- *Mudança no local, horário ou data do evento.*

**RF14**– *Deletar evento.* **Informações:** *O administrador poderá deletar algum determinado evento.* **Regras:** *O administrador poderá deletar os eventos nos seguintes caso:*

- *O evento não estar relacionado a esports;*
- *O evento não for devidamente legalizado.*

**RF15**– *Bloquear usuários.* **Informações:** *O administrador poderá bloquear alguns usuários.* **Regras:** *Os casos de bloqueio são:*

- *Usuário apresentar informações falsas;*
- *Usuário apresentar reports inexistentes.*
- *Usuário divulgador criar eventos inexistentes.*

**RF16**– *Gerar ingresso.* **Informações:** *O sistema poderá gerar ingressos de um determinado evento, em caso de compras.* **Regras:** *O ingresso deve conter os seguintes elementos:*

- *Código identificador;*
- *Local (cadeira, poltrona ou seção);*
- *Data de compra;*
- *Data do evento com uma pequena descrição;*
- *Marca d'água própria do sistema.*

## 5.2 Requisitos Não Funcionais

**RNF01**– O backend deve ser implementado usando NodeJs. **Informações:** O código do backend deve ser feito usando javascript ou typescript.

**RNF02**– O banco de dados utilizado deve ser MongoDB. **Informações:** É um banco de dados orientado a documentos livres. **Regras:** Deve ser usado mongoAtlas.

**RNF03**– Reembolso em caso de cancelamento do evento. **Informações:** Reebolso de ingresso. **Regras:** Para que um evento seja cancelado deve-se reembolsar todos que já compraram o ingresso daquele evento.

**RNF04**– Reembolso tem que ser feito com antecedência. **Informações:** Reebolso de ingresso. **Regras:** Para realizar o reembolso de um ingresso comprado, é necessário que seja feito com no mínimo 2 dias de antecedência ou não será possível fazê-lo.

- O sistema deve ser fazer a validação dos dados.

**RNF04**– O valor do evento não pode ser modificado **Informações:** Não é possível uma atualização no valor do ingresso. **Regras:** Após publicar um evento não é permitido que altere o valor do ingresso.

- O sistema deve ser fazer a validação dos dados.

**RNF05**– Deve ter um número máximo de pessoas por evento. **Informações:** Os locais tem capacidade máxima. **Regras:** O número de ingressos vendidos não deve superar a lotação máxima do local do evento.

- O sistema deve ser fazer a validação dos dados.

**RNF07**– Os eventos tem idade mínima. **Informações:** Todos os eventos devem especificar a idade mínima. **Regras:** Um evento obrigatoriamente deve definir uma faixa etária mínima para compra dos ingressos.

- O sistema deve ser fazer a validação dos dados.

**RNF08**– Ingresso deve respeitar a faixa etária mínima. **Informações:** Os dados do usuário incluem idade. **Regras:** Não é permitido venda de ingresso para usuário que a idade seja menor do que a idade mínima do evento.

- O sistema deve ser fazer a validação dos dados.



## **6 Serviços**

### **6.1 Serviço de usuário**

*Serviços destinados aos usuários padrão.*

- *RF01*
- *RF02*
- *RF03*
- *RF05*
- *RF06*
- *RF07*
- *RF08*

### **6.2 Serviço de gerenciamento de eventos**

*Serviços destinados aos usuários organizadores dos eventos.*

- *RF09*
- *RF10*
- *RF11*
- *RF12*

### **6.3 Serviço de Administrador**

*Serviços destinados aos usuários administradores do sistema.*

- *RF04*
- *RF13*
- *RF14*
- *RF15*

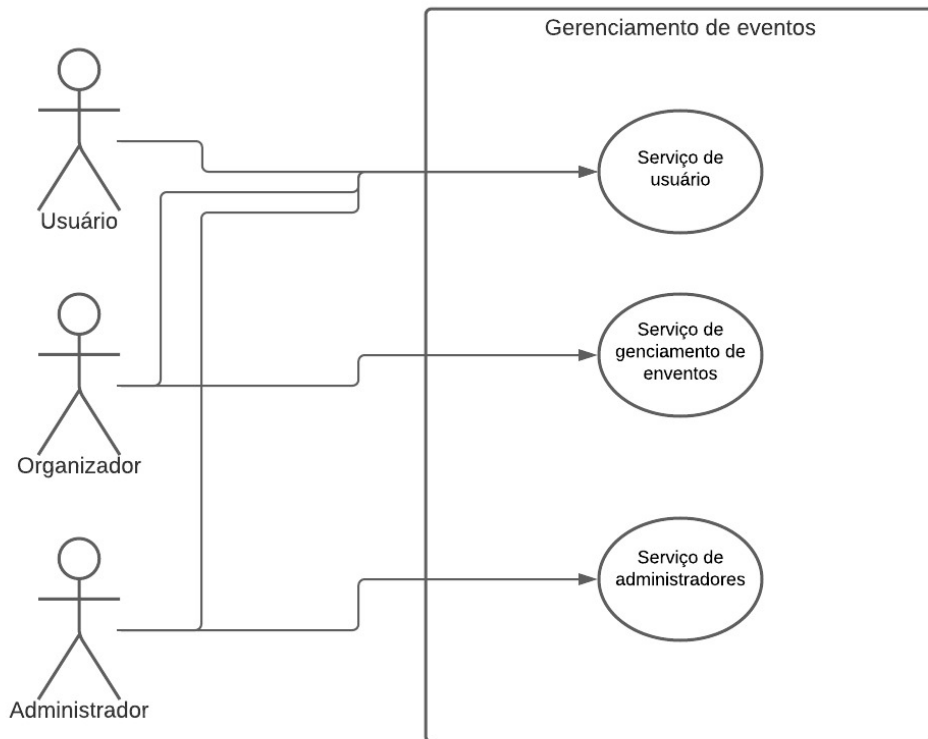


Figura 1: Diagrama de casos de uso

## 7 Plano de VVT

*Esta parte do documento visa explicitar os testes implementados no sistema. Em suma serão testes de integração que deverão ser executados automaticamente numa pipeline de produção e desenvolvimento com objetivo de manter a integridade do software.*

*Será descrito, também, os requisitos para execução dos teste e a descrição da equipe de desenvolvimento desses testes.*

### 7.1 Requisitos a serem testados

*O plano de testes foi dividido nos seguintes campos:*

- *Teste das regras de negócio*
- *Testes interface*
- *Testes de segurança*

- *Teste de desempenho*

## 7.2 Estratégias e ferramentas de teste

*Os teste de validação da regra de negócio serão desenvolvidos seguindo a metodologia de testes caixa preta onde acontecerá a validação por meio da interface do usuário, juntamente com os testes de interface, para esse tipo de teste será utilizadas as ferramentas Junit com Cucumber escrita em linguagem Gherkin muito presente no BDD (Behavior Driven Development). O objetivo do uso da linguagem Ghekin é facilitar a leitura e escrita das regras de negócio para o cliente e product owner.*

## 7.3 Equipe e infra-estrutura

*Como citado anteriormente a equipe de infra-estrutura e testes devem implementar uma pipeline para automatização dos testes tanto quanto o ambiente de homologação e produção para diferenciação de código já disponibilizado e ainda em desenvolvimento.*

## 7.4 Execução do Plano de Teste

- ***Reembolso em caso de cancelamento do evento..***
  - ***Objetivo:*** *Verificar e garantir que o sistema obedeça a regra de negócio.*
  - ***Critério de aceitação:*** *Dado o cenário de teste o sistema não deve permitir o usuário cancelar o evento com devoluções pendentes.*
- ***Reembolso tem que ser feito com antecedência.***
  - ***Objetivo:*** *Verificar e garantir que o sistema obedeça a regra de negócio.*
  - ***Critério de aceitação:*** *Dado o cenário de teste o sistema não deve permitir o usuário reembolse seu ingresso de um evento com um prazo menor do que dois dias úteis.*
- ***O valor do evento não pode ser modificado.***
  - ***Objetivo:*** *Verificar e garantir que o sistema obedeça a regra de negócio.*

- **Critério de aceitação:** Dado o cenário de teste o sistema não deve permitir a alteração do valor de um evento cadastrado.
- **Deve ter um número máximo de pessoas por evento.**
  - **Objetivo:** Verificar e garantir que o sistema obedeça a regra de negócio.
  - **Critério de aceitação:** Dado o cenário de teste o sistema não deve permitir a venda de um número maior de ingressos do que o campo "número máximo de pessoas" cadastrado no evento.
- **Os eventos tem idade mínima.**
  - **Objetivo:** Verificar e garantir que o sistema obedeça a regra de negócio.
  - **Critério de aceitação:** Dado o cenário de teste o sistema deve garantir que o campo idade mínima seja cadastrado junto com um evento.
- **Ingresso deve respeitar a faixa etária mínima..**
  - **Objetivo:** Verificar e garantir que o sistema obedeça a regra de negócio.
  - **Critério de aceitação:** Dado o cenário de teste o sistema deve garantir que a idade do usuários seja menor que a da faixa etária do evento.
- **Verificar cadastro de usuário.**
  - **Objetivo:** Verificar que um único usuário foi cadastrado.
  - **Critério de aceitação:** Dado que o usuário preencheu o formulário de cadastro e o enviou, deve verificar no banco que existe apenas uma entidade com os mesmos dados cadastrados.
- **Verificar cadastro de evento.**
  - **Objetivo:** Verificar que um único evento foi cadastrado.
  - **Critério de aceitação:** Dado que o usuário preencheu o formulário de cadastro de evento e o enviou, deve verificar no banco que existe apenas uma entidade com os mesmos dados cadastrados.
- **Testes de segurança**

- *Teste de desempenho*

<b>Funcionalidade:</b> Envelope		<b>Tempo Despendido(h):</b> 1h
<b>Contador:</b> 02		<b>Criticidade:</b> Baixa
<b>Objeto de Teste:</b> <ul style="list-style-type: none"> <li>➤ Validar uso do assistente para criar documento no modelo Envelope.</li> <li>➤ Verificar impressão usando um envelope.</li> </ul>		
<b>Descrição do Caso de Teste:</b> <ul style="list-style-type: none"> <li>➤ O aplicativo deverá atender corretamente a configuração fornecida pelo usuário (nas 3 abas: Envelope, Formato e Impressora).</li> <li>➤ A impressão no envelope deverá ser realizada com sucesso.</li> </ul>		
<b>Pré-Condição:</b> <ul style="list-style-type: none"> <li>➤ O usuário deverá informar dados do Destinatário e Remetente.</li> <li>➤ Existir uma impressora compatível para impressão no envelope.</li> </ul>		
<b>Dados de Entrada:</b>		
ID	Passo	Procedimento
1	P1	Executar o aplicativo LibreOffice 4.2 (Opção texto)
	P2	Selecionar a opção Texto
	P3	Acessar o menu a opção Envelope (Inserir>>Envelope)
	V1	O aplicativo deverá exibir a tela do assistente para Envelope
	P3	Preencher os dados do campo Destinatário e Remetente
	P4	Clicar no botão Inserir
2	P5	Se necessário ajustar a largura da caixa de texto do Destinatário e Remetente
	P7	Clicar no ícone da Impressora
	V2	Verificar se o envelope foi impresso corretamente.
<b>Resultado Esperado:</b> As operações deverão funcionar corretamente cumprindo todas as regras acima citadas.		

Figura 2: Exemplo

## 8 Medição e Qualidade de Software

Para realizar medições e qualidade de software serão realizados umas sequências de testes em todo o sistema. Para o frontend terão testes simulam o comportamento de um usuário no sistema e também testes de desempenho, utilizando o tempo de execução como métrica. Para o backend será feitos teste que simulam as requisições de um usuário utilizando o sistema e também terá testes de desempenho baseados no tempo, desconsiderando problemas de rede, todas as chamadas devem ter um tempo maximo de dez segundos para enviar a resposta de volta para o usuário.

## 9 Observações

*Apresente aqui as dificuldades na disciplina, trabalho prático e coisas do tipo.*

## 10 Referências

[1] *<https://scrumguides.org/scrum-guide.html>*;