

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 Willian Gomes |
| Professor: | Msc prof. Johnatan Oliveira |

Horário: Seg & Qua - 08:20 -10:00

Ouro Preto, 06 de Março de 2021

Conteúdo

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

1 Histórico de Revisões

| Data | Versão | Descrição | Autor |
|------------|--------|--|-------|
| 12/03/2021 | 0.0 | Levantamento de requisitos | PCC |
| 16/03/2021 | 0.0 | Descrição dos requisitos funcionais do cliente | PCC |
| | | | |

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 1 | Escrever o plano de testes | 01/01 a 10/01 |
| Fulano 2 | Testar a ferramenta proposta no plano de testes | 01/01 a 05/01 |
| | | |

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 divulgador 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 divulgador, 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 divulgador 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 casos:*

- *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

RFN01– O backend deve ser implementado usando NodeJs. **Informações:** O código do backend deve ser feito usando javascript ou typescript.

RFN02– O banco de dados utilizado deve ser MongoDB. **Informações:** É um banco de dados orientado a documentos livres. **Regras:** Deve ser usado mongoAtlas.

RFN03– 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.

RFN04– Reembolso em caso de cancelamento de compra. **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.

RFN05– 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.

RFN06– 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.

RFN07– 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.

RFN08– Os eventos tem idade mínima. **Informações:** Todos os eventos devem especificar a idade mínima. **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

- *RF01*
- *RF02*
- *RF03*
- *RF06*
- *RF07*

6.2 Serviço de gerenciamento de eventos

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

7 Plano de VVT

Asseguram que o software cumpra com suas especificações e atenda às necessidades dos usuários. Você deve apresentar um plano de testes, ferramentas que serão utilizadas e coisas do tipo.

Veja um exemplo no link: https://www.cin.ufpe.br/~gta/rup-vc/extend.formal_resources/guidances/examples/resources/test_plan_v1.htm

7.1 Requisitos a serem testados

Esta seção descreve em linhas gerais o conjunto de requisitos a serem testados no projeto a ser desenvolvido, comunicando o que deve ser verificado. Exemplos de requisitos a serem testados são: desempenho, segurança, interface de usuário, controle de acesso, funcionalidades.

7.2 Estratégias e ferramentas de teste

Apresenta um conjunto de tipos de testes a serem realizados, respectivas técnicas empregadas e critério de finalização de teste. Além disso, é listado o conjunto de ferramentas utilizadas.

7.3 Equipe e infra-estrutura

Contém descrição da equipe e da infra-estrutura utilizada para o desenvolvimento das atividades de testes, incluindo: pessoal, equipamentos, software de apoio, materiais, dentre outros. Isto visa garantir uma estrutura adequada para a execução das atividades de testes previstas no plano.

7.4 Execução do Plano de Teste

| | |
|--------------------------|-------------------------|
| Funcionalidade: Envelope | Tempo Despendido(h): 1h |
| Contador: 02 | Criticidade: Baixa |

| | | |
|--|--------------|---|
| Objeto de Teste: | | |
| <ul style="list-style-type: none">➤ Validar uso do assistente para criar documento no modelo Envelope.➤ Verificar impressão usando um envelope. | | |
| Descrição do Caso de Teste: | | |
| <ul style="list-style-type: none">➤ O aplicativo deverá atender corretamente a configuração fornecida pelo usuário (nas 3 abas: Envelope, Formato e Impressora).➤ A impressão no envelope deverá ser realizada com sucesso. | | |
| Pré-Condição: | | |
| <ul style="list-style-type: none">➤ O usuário deverá informar dados do Destinatário e Remetente.➤ Existir uma impressora compatível para impressão no envelope. | | |
| Dados de Entrada: | | |
| 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. |
| | | |

| |
|---|
| Resultado Esperado: As operações deverão funcionar corretamente cumprindo todas as regras acima citadas. |
|---|

Figura 1: Exemplo

8 Medição e Qualidade de Software

Apresente aqui o formato da Medição e qualidade de software. Você deve mostrar os meios que irá avaliar a qualidade do seu software. Apresente o

plano e os resultados a partir da prática de ferramentas de detecção de code smells, por exemplo. Em Java, temos uma ferramenta chamada JDEODORANT. Você pode avaliar as métricas de qualidade também, por exemplo, em Java, temos CKMetrics¹

9 Observações

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

¹<https://github.com/mauricioaniche/ck>

10 Referências

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