

Sistema de Votação do Restaurante Universitário

Firmware

Plano de Teste

Versão 1.0

Histórico da Revisão

Data	Versão	Descrição	Autor
10/06/2018	1.0	Teste da primeira versão do sistema completamente integrado	Pedro Victor

Sumário

1. INTRODUÇÃO	4
1.1 Escopo	5
1.2 Não Escopo	5
1.3 Acrônimos e terminologias do documento	5
2. PLANEJAMENTO PARA OS TESTES	6
2.1 Necessidades de Hardware	6
2.2 Necessidades de Software	7
2.3 Necessidade de Pessoas	7
2.4 Necessidade de Capacitação	7
2.5 Cronograma de Testes	8
2.6 Riscos e Restrições	8
3. TESTE DE UNIDADE	
4. TESTE DE INTEGRAÇÃO	9
5. TESTE DE TRANSIÇÃO DE ESTADOS	

1. INTRODUÇÃO

1.1 Escopo

O Firmware do Sistema de Votação do Restaurante Universitário possui poucos casos de uso. Dessa forma, serão necessários somente os tipos de testes de unidade, de transição de estados e de integração.

- No Teste de Unidade verificaremos os componentes separadamente. Os componentes são as variáveis de ambiente e algumas rotinas que operam no firmware do sistema.
- No Teste de Transição de Estados definiremos cada estado para posteriormente mostrar como as variáveis lidam com os diversos tipos de transições possíveis no autômato finito do firmware.
- No Teste de Integração temos como objetivo encontrar erros na integração de unidades e de componentes. É um teste mais difícil, pois exige que o testador conheça muito bem as relações existentes no código.

1.2 Não Escopo

Teremos dois tipos de testes que não farão parte do planejamento de verificação do firmware. O primeiro teste se chama teste de sistema e o segundo se chama teste de aceitação.

- Teste de Sistema: É um teste complexo que permite verificar a integração entre componentes de software e de hardware. Percebe-se que é um teste que não é muito viável, pois se trata de avaliar partes funcionais do código relacionadas ao hardware
- Teste de Aceitação: Basicamente, permite avaliar se o software do sistema funcionará de maneira esperada. Para avaliar a corretude do software é necessário que a equipe de teste se atente às especificações de requisitos funcionais e não funcionais.

1.3 Acrônimos e terminologias do documento

Esse ponto descreve os termos e abreviaturas importantes que serão mencionados no decorrer deste documento. Estes termos são descritos na Tabela 1, estando apresentados por ordem alfabética.

SVRU	Sistema de Votação do Restaurante Universitário
IDE	Integrated Development Environment

2. PLANEJAMENTO PARA OS TESTES

2.1 Necessidades de Hardware

Tipo de Hardware	Detalhamento	Quantidade	Forma de Disponibilização	Data Limite
Microcontrolador ATmega 328p	Microcontrolador utilizado para receber o firmware	1	Corporativo	
Módulo Bluetooth HC-06	Dispositivos que fará a comunicação com o dispositivo móvel	1	Corporativo	

2.2 Necessidades de Software

Tipo de Software	Detalhamento	Quantidade	Forma de Disponibilização	Data Limite
Eclipse Oxygen	Versão 4.7	1	Corporativo	
ATmel Studio 7	Versão 7.0.1645	1	Corporativo	
CUnit	Versão 2.1.3	1	Corporativo	
Yakindu	Versão 3.3.0	1	Corporativo	

2.3 Necessidade de Pessoas

Papel	Envolvimento Estimado	Quantidade	Período de Envolvimento no Projeto
Equipe de Teste: Pedro Victor	2 horas	1	

2.4 Necessidade de Capacitação

Treinamento	Duração	Data de Realização
Ter habilidade com os programas de testes e com o firmware.	2 horas	06/06/2018

2.5 Cronograma de Testes

Atividade	Data de Início	Duração (horas)	Papel Responsável/Envolvidos
Teste Unitário	10/06/2018	2 horas	Pedro Victor
Teste de Integração	12/06/2018	3 horas	Pedro Victor
Teste de Transição de Estados	15/06/2018	2 horas	Pedro Victor

2.6 Riscos e Restrições

Riscos: A equipe de teste não possui conhecimento necessário para usar as ferramentas.

Restrições: Não há restrições de testes no firmware.

3. Teste de Unidade

Este tipo de teste será feito individualmente em cada unidade, onde cada unidade será uma função. A ferramenta que iremos utilizar será a IDE Eclipse Oxygen junto do Framework CUnit. Este último permitirá fazer os testes automatizados.

O teste funcionará da seguinte forma:

- 1 - Definir a função que será testada.
- 2 - Criar um banco de testes utilizando a ferramenta CUnit. Cada função de teste será executada e avaliará a funcionalidade de uma determinada função. A melhor maneira de criar esse banco de testes é elaborar uma biblioteca de funções de teste.
- 3 - Executar imediatamente as funções de teste para cada funcionalidade criada.

A melhor forma de usar esse teste é saber qual funcionalidade será criada e , posteriormente, elaborar as funções de teste. Isto é útil, pois permite que a propagação de erros seja menor.

4. Teste de Integração

O teste de integração será feito com a IDE Eclipse e com o Framework CUnit. Esse teste se assemelha bastante ao teste de unidade, no entanto este testará as relações de diversas funções que envolvem requisitos funcionais. Para fazer este teste é melhor que o teste de unidade tenha sido feito anteriormente, pois isso garantirá que as funcionalidades estejam corretas pelo menos de forma independente. Para realizá-lo usaremos os documentos que apresentam os requisitos funcionais e o diagrama de blocos.

O teste funcionará da seguinte forma:

- 1 - Analisar o diagrama de blocos e observar as relações entre as funcionalidades.
- 2 - Para cada relação criar um banco de testes com o CUnit. Dessa forma, será possível que as funções de teste avaliem as funcionalidades que têm dependências.

Novamente, este teste será mais eficaz se a criação dessas dependências forem criadas simultaneamente ao banco de testes referente à relação.

5. Teste de transição de estados

Caso de teste	Transição	Estado origem	Entrada	Saída	Estado destino	Saída Obtida
Inicia	Inicia >> Refeição Inicia >> Qualidade Inicia >> Erro	Inicia Votação	Variáveis de ambiente reiniciadas em zero	Permissão para votar	Refeição/Qualidade/Erro	Permissão para votar
Refeição	Refeição >> Valida Voto Refeição >> Erro	Escolhe Refeição	Cvermelha/ Cbranca/ Vegetariano	Variável relacionada à refeição setada	Valida Voto/ Erro	Acende led relacionado à refeição
Qualidade	Qualidade >> Valida Voto Qualidade >> Erro	Escolhe Qualidade	Otimo/ Regular/ Ruim	Variável relacionada à qualidade setada	Valida Voto/ Erro	Acende led relacionado à qualidade
Erro	Erro >> Inicia	Estado de Erro	pressConfirma	Após 5 segundos: Ir para o estado de início	Inicia	
Valida Voto	Valida Voto >> Valida Voto Valida Voto >> Confirma	Valida Voto	Cvermelha/ Cbranca/ Vegetariano / Otimo/ Regular/ Ruim	Variáveis de ambientes setadas conforme a escolha do usuário	Valida Voto/ Confirma	Acende led relacionado à refeição ou à qualidade
Confirma	Confirma >> Inicio	Confirma	pressConfirma	Após 1 segundo iniciará a votação	Inicia	Acende led de confirmação

