

# **Sistema de Votação do Restaurante Universitário** **Software**

## **Plano de Teste**

**Versão 1.0**

## Histórico da Revisão

DATA	VERSÃO	DESCRIÇÃO	AUTOR
<26/05/2018>	<1.0>	Teste da primeira versão do sistema completamente integrado	Ruan Derlan Sombra Oliveira

# Sumário

## Sumário

1.	INTRODUÇÃO.....	4
1.1	Escopo.....	4
1.2	Não Escopo.....	4
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.....	6
2.3	Necessidade de Pessoas.....	6
2.4	Necessidade de Capacitação.....	6
2.5	Cronograma de Testes.....	7
2.6	Riscos e Restrições.....	7
3.	Teste de transição de estados.....	8
4.	Teste de Sistema.....	9

## 1. INTRODUÇÃO

Este documento de Plano de Teste tem o objetivo de documentar as informações necessárias para planejar e controlar os testes de validação do projeto SVRU. O documento descreve o plano geral de testes referente aos cadastros básicos de forma a direcionar os esforços de teste, teste de transição de estados e os Casos de Teste a serem executados para validar o produto.

Além de achar falhas, testes visam expandir a confiabilidade de um sistema, ou seja, aumentar a chance de que o sistema seguirá funcionando sem apresentar erros pelo máximo de tempo possível sem que apresente falhas. Teste de Software é uma das atividades do processo de construção de sistema de software que tende executar um programa de modo sistemático com o propósito de constatar falhas que podem estar presentes no sistema.

### 1.1 Escopo

Sendo assim é necessário verificar e validar o software em questão, definindo dessa forma quando as funções de verificação e validação iniciam e terminam. Para Software do Sistema de Votação do Restaurante Universitário, serão utilizados apenas os testes de sistema, aceitação e de transição de estados.

- 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.
- 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 software.
- Teste de Sistema: embora complexo, permite verificar a integração entre componentes de software e de hardware. Mesmo que seja considerado um teste que não é muito viável, pois se trata de avaliar partes funcionais do código relacionadas ao hardware, é necessário e eficaz.

### 1.2 Não Escopo

Teremos dois tipos de testes que não abordaremos para o planejamento de verificação de erros referentes ao software. O primeiro é o teste de unidade e o outro se chama teste de integração.

- No Teste de Unidade verifica-se os componentes de forma separada. Os componentes são as variáveis de ambiente e algumas rotinas que operam no firmware do sistema.
- 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.

### 1.3 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 a seguir, estando apresentados por ordem alfabética.

App Inventor for Android	Aplicação código aberto originalmente criada pela Google
SVRU	Sistema de Votação do Restaurante Universitário

## 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 a criação do sistema	1	Corporativo	
Módulo Bluetooth HC-06	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	
MIT AI2 Companion	Versão 2.46	1	Corporativo	

### 2.3 Necessidade de Pessoas

PAPEL	ENVOLVIMENTO ESTIMADO	QUANTIDADE	PERÍODO DE ENVOLVIMENTO NO PROJETO
Testador	5 horas	1	10/06/2018 à 15/06/2018

### 2.4 Necessidade de Capacitação

TREINAMENTO	DURAÇÃO	DATA DE REALIZAÇÃO
Ter conhecimento sobre desenvolvimento de software no App Inventor for Android, habilidade com testes de programas.	5 horas	06/06/2018

## 2.5 Cronograma de Testes

### TESTES DE SISTEMA

ATIVIDADE	DATA DE INÍCIO	DURAÇÃO (HORAS)	PAPEL RESPONSÁVEL/ENVOLVIDOS
Teste de Aceitação	10/06/2018	3 horas	Ruan Derlan
Teste de Transição de Estados	12/06/2018	2 horas	Ruan Derlan e Camila Stéfany
Teste de Sistema	15/06/2018	4 horas	Ruan Derlan e Camila Stéfany

## 2.6 Riscos e Restrições

Equipe de testes não tem domínio da ferramenta de testes de carga.

Erro na conexão com Bluetooth.

Não seguir a sequência correta dos botões.

### 3. TESTE DE TRANSIÇÃO DE ESTADOS

Caso de teste	Transição	Estado origem	Entrada	Saída	Estado destino	Saída Obtida
teste de opções	opção -> verificação bluetooth	opção	confirmar opções	opções confirmadas	verificação bluetooth	itens selecionados
teste de conexão bluetooth	verificação bluetooth -> armazenamento de dados	verificação bluetooth	verificação da conexão bluetooth	conexão estabelecida	armazenamento de dados	ocorreu conexão via bluetooth
teste de armazenamento	armazenamento de dados -> início	armazenamento de dados	armazenar dados	dados armazenados	início	dados foram armazenados no banco de dados
teste de início	início - > opção	início	escolher resgate de dados	tela selecionar opções	opção	-



#### 4. TESTE DE SISTEMA

<b>ID:</b>	CT001
<b>Caso de Teste:</b>	Enviar dados
<b>Requisito Funcional associado:</b>	-
<b>Pré-Condição:</b>	1. O aparelho que realizará o recebimento de dados deve ter o aplicativo SVRU instalado e o mesmo deve estar iniciado e com a conexão de Bluetooth ativa.
<b>Procedimento:</b>	1. O SVRU Firmware inicia a comunicação do o SVRU Firmware via Bluetooth e faz uma solicitação para saber se pode enviar dados ao SVRU Software. 2. O SVRU Software retorna uma confirmação, onde afirma estar pronto para receber os dados. 3. O SVRU Firmware gera um breve relatório de informações e envia para o SVRU Software.
<b>Resultado Esperado:</b>	1. Envio dos dados seja realizado com sucesso.
<b>Resultado obtido:</b>	

<b>ID:</b>	CT002
<b>Caso de Teste:</b>	Coletar dados do SVRU Firmware
<b>Requisito Funcional associado:</b>	-
<b>Pré-Condição:</b>	1. Os dados enviados pelo SVRU Firmware serem recebidos corretamente.
<b>Procedimento:</b>	1. Usuário escolhe a opção de Coletar dados 2. Sistema mostra a opção para digitar data 3. Usuário insere a data desejada 4. Sistema mostra a opção de refeição. 5. Usuário seleciona a opção desejada entre: almoço ou jantar. 5. Sistema verifica conexão Bluetooth. 6. Sistema coleta os dados
<b>Resultado Esperado:</b>	1. Uma janela de validação, exibindo uma mensagem para sinalizar que a coleta dos dados foi realizada com sucesso.
<b>Resultado obtido:</b>	

<b>ID:</b>	CT003
<b>Caso de Teste:</b>	Armazenar dados
<b>Requisito Funcional associado:</b>	-
<b>Pré-Condição:</b>	1. A coleta de dados tenha sido bem sucedida.
<b>Procedimento:</b>	1. Sistema mostra os dados coletados. 2. Sistema armazena dados de acordo com sua data e tipo de refeição. 3. Sistema organiza os armazenamentos em forma de uma lista ordenada por datas.
<b>Resultado Esperado:</b>	1. Armazenamento de dados seja realizado com sucesso.
<b>Resultado obtido:</b>	