

---

---

## **Plano de Testes**

Desenvolvimento do Sistema de Diário Eletrônico

**Professora:**

Gláucia Braga

**Equipe:**

Elena Augusta Araujo

Augusto Araujo Domingues

Lucas Burahem

Délio Costa

**2015**

**Histórico de Revisões**

<b>Data</b>	<b>Versão</b>	<b>Descrição</b>	<b>Autor</b>
<10/setembro/15>	<1.0>	Release Inicial	Augusto / Elena
<23/setembro/15	<1.1>	Adição de casos de teste	Augusto / Elena

# **Índice**

## **1. IDENTIFICADOR DO PLANO DE TESTE**

## **2. INTRODUÇÃO**

### **2.1 OBJETIVOS**

### **2.2 ESCOPO**

## **3. ITENS DE TESTE**

### **3.1 REQUISITOS A SEREM TESTADOS**

### **3.2 FUNCIONALIDADES A SEREM TESTADAS**

#### **3.2.1 TESTE DE BANCO DE DADOS**

#### **3.2.2 TESTE FUNCIONAL**

#### **3.2.3 TESTE DA INTERFACE DE USUÁRIO**

#### **3.2.4 TESTE DE CONTROLE DE ACESSO**

### **3.3 FUNCIONALIDADES QUE NÃO SERÃO TESTADAS**

### **3.4 CRITÉRIOS DE ENTRADA E SAÍDA**

## **4. ABORDAGEM DE TESTE**

## **5. TAREFAS DE TESTE**

## **6. RESPONSABILIDADES**

## **7. CRONOGRAMA DE TESTES**

## **8. APROVAÇÕES**

## 1. Identificação de Projeto

Este projeto consiste em promover um trabalho interdisciplinar entre quatro disciplinas do curso de Ciência da Computação no que compete ao desenvolvimento de um protótipo de software para um sistema de controle de frequência da UFV - campus Florestal.

Cada disciplina é responsável por abordar suas respectivas áreas. A Engenharia de Software II controla o gerenciamento do projeto, a elaboração dos cronogramas, realiza testes, prototipa interface dentre outros aspectos gerenciais. Em Arquitetura de software I, são definidas as visões lógicas e físicas da arquitetura do sistema, recomendações sobre o uso de padrões arquiteturais e de design, revisão dos modelos de classes, de dados e implementação da integração entre camadas. Banco de Dados I, tem por finalidade elaborar os modelos lógico e físico do banco de dados, implementar o banco de dados no SGBD MySQL, inserir dados para algumas tabelas de forma direta e elaborar as consultas em SQL para atender às demandas de programação. Finalmente, Programação Orientada a Objetos, tem por responsabilidade modelar as classes e implementar as camadas de modelo e controle, segundo o paradigma da orientação a objetos.

A oportunidade de desenvolver este trabalho proporcionará aos alunos uma visão de como realmente é o processo de desenvolvimento de um software e de que maneira se deve ocorrer executando-o de forma colaborativa.

A tabela abaixo identifica a documentação e disponibilidade usados para desenvolver o plano de testes:

Documento	Criado ou Disponível	Recebido ou Revisado
Especificação de Requisitos	■ Sim □ Não	■ Sim □ Não
Documento Arquitetural	■ Sim □ Não	■ Sim □ Não
Protótipo	■ Sim □ Não	■ Sim □ Não

## 2. Introdução

Este projeto objetiva promover um trabalho interdisciplinar entre quatro disciplinas do curso de Ciência da Computação no que compete ao desenvolvimento de um protótipo de software para um sistema de controle de frequência da UFV - campus Florestal.

### 2.1 Objetivos

Esse documento do Plano de Testes compõe-se dos seguintes objetivos:

- Identificar informações de projeto existentes e os componentes de software que devem ser testados.
- Listar os Requisitos a Testar recomendados (alto nível).
- Recomendar e descrever as estratégias de teste a serem empregadas.
- Identificar os recursos necessários e prover uma estimativa dos esforços de teste.
- Listar os elementos resultantes do projeto de testes.

## **2.2 Escopo**

O sistema desenvolvido para controle de frequência da UFV passará pelos testes funcionais, de banco de dados e de interface com o usuário.

## **3 Itens de teste**

### **3.1 Requisitos a serem testados**

Este documento identifica os requisitos funcionais do sistema de controle de frequência da UFV. Estes requisitos foram elaborados de acordo com as especificações necessárias para a primeira etapa do desenvolvimento deste sistema.

**RF01** - O sistema deverá ser capaz de obter dados de cursos, disciplinas, turmas, datas, horários, locais, semestre letivo, professores e alunos.

**RF01.1** - Como alternativa provisória à integração com SAPIENS, esses dados deverão ser inseridos previamente no banco de dados.

**RF01.2** - O sistema deverá armazenar dados de cursos, contendo os campos: nome, sigla, grade curricular (disciplinas de um curso), carga horária, tipo (presencial ou à distância).

**RF01.3** - O sistema deverá armazenar dados de disciplinas, contendo os campos: nome, sigla, carga horária total (carga horária teórica + carga horária prática, tipo (teórica, prática ou teórico-prática).

**RF01.4** - O sistema deverá armazenar dados de turmas de disciplinas, contendo os campos: nome (exemplo: CCF323–ARQUITETURA DE SOFTWARE - Turma Teórica 1), professor responsável, horários, locais, alunos e datas das aulas.

**RF02** - O sistema deverá realizar o controle de acesso, considerando dois tipos de usuários: docentes e discentes.

**RF02.1** – O sistema deverá efetuar o controle de acesso, por meio da validação de usuário (matrícula) e senha .

**RF03** - O sistema deverá configurar uma tela com o diário eletrônico, conforme o protótipo mostrado na Figura 3, de acordo com a disciplina selecionada pelo docente.

**RF03.1** - O sistema deverá ser capaz de selecionar disciplinas de um determinado curso;

**RF03.1.1** - O sistema deverá ser capaz de selecionar turmas de uma determinada disciplina;

**RF03.1.1.1** – O sistema deverá ser capaz de selecionar alunos de uma turma e exibi-los em ordem alfabética.

**RF04** – O sistema deverá permitir ao docente o lançamento do calendário acadêmico de cada turma de disciplina, contendo as datas das aulas correspondentes.

**RF04.1** – O sistema deverá permitir o lançamento do conteúdo ministrado em cada aula, para cada data lançada.

**RF04.2** – O sistema deverá ser capaz de distribuir as aulas, segundo o calendário acadêmico de cada turma, ao longo do semestre.

**RF04.3** – O sistema deverá ser capaz permitir o lançamento de aulas extras pelo docente.

**RF05** - O sistema deverá ser capaz de registrar as presenças/ausências dos alunos para cada aula no cronograma da turma, clicando em cada célula correspondente. Por padrão, o aluno recebe presença e diante de um clique de mouse, o valor da célula passa a conter um 'F', indicando falta do aluno.

**RF05.1** – O sistema deverá calcular e registrar o total de faltas para cada aluno de uma turma, por solicitação do docente ou ao final de cada mês do semestre letivo.

**RF05.1.1** – O sistema deverá interoperar com o sistema SAPIENS, ao final de cada mês para atualizar essa informação.

**RF06** - O sistema deverá ser capaz de gerar relatório de faltas acumuladas por turma de cada disciplina ministrada por um docente.

**RF07** - O sistema deverá ser capaz de gerar um relatório de faltas acumuladas para todas as turmas de disciplinas nas quais o aluno está matriculado.

**RF07.1** – O sistema deverá ser capaz de gerar um relatório resumido, exibindo o total de faltas para cada turma de disciplina na qual o aluno está matriculado.

**RF07.2** – O sistema deverá ser capaz de gerar um relatório detalhado, exibindo as datas das faltas (dia/mês/ano) para cada turma de disciplina na qual o aluno está matriculado ao final de cada mês.

### **3.2 Funcionalidades a serem testadas**

As funcionalidades a serem testadas neste documento condizem com os requisitos funcionais do projeto. Foram abordados os testes de banco de dados, testes

funcionais, teste de interface de usuário e teste de controle de acesso.

### 3.2.1 Teste do Banco de Dados

**TB-01:** Verificar se as informações de alunos podem ser cadastradas, consultadas e removidas.

**TB-02:** Verificar se as informações dos cursos podem ser cadastradas, consultadas e removidas.

**TB-03:** Verificar se as informações das disciplinas podem ser cadastradas, consultadas e removidas.

**TB-04:** Verificar se as informações de turmas de disciplinas podem ser cadastrados, consultados e removidos.

**TB-05:** Verificar se as informações de um usuário (discentes/ docentes) podem ser cadastradas, consultadas e removidas.

### 3.2.2 Teste Funcional

**TF-01:** Verificar que qualquer usuário pode acessar sua própria conta através de login e senha.

#### Critérios de entrada e saída

<u>Entrada</u>	<u>Saída Esperada</u>
Usuário válido - Senha válida	Login efetuado com sucesso
Usuário inválido - Senha inválida	Erro no login ("Usuário ou senha inválidos")
Usuário válido - Senha inválida	Erro no login ("Usuário ou senha inválidos")
Usuário inválido - Senha válida	Erro no login ("Usuário ou senha inválidos")
Usuário nulo - Senha qualquer	Erro no login ("Preencha todos os campos")
Usuário nulo - Senha nula	Erro no login ("Preencha todos os campos")

**TF-02:** Verificar a capacidade do sistema de selecionar as disciplinas de um determinado curso.

<u>Entrada</u>	<u>Saída Esperada</u>
Clique em determinado curso apresentado	Retorno das disciplinas que compõe um determinado curso.

Cadastrar um novo curso	Array de disciplinas deve retornar vazio
-------------------------	--

**TF-03:** Verificar se o sistema é capaz de selecionar as turmas de uma determinada disciplina.

<u>Entrada</u>	<u>Saída Esperada</u>
Clique em determinada disciplina apresentada	Retorno das turmas aquela disciplina possui.

**TF-04:** Verificar se o sistema é capaz de selecionar os alunos de uma turma e exibi-los em ordem alfabética.

<u>Entrada</u>	<u>Saída Esperada</u>
Clique em determinada turma apresentada	Retorno dos alunos matriculados na turma em ordem alfabética

**TF-05:** Verificar se o sistema é capaz de acessar o cronograma de uma determinada turma.

<u>Entrada</u>	<u>Saída Esperada</u>
Selecionar uma determinada turma	O sistema exibir o cronograma daquela determinada turma, com os dados do seu tipo de carga horário, professor responsável.

**TF-06:** Verificar se o sistema é capaz de permitir o lançamento do conteúdo ministrado em cada aula, para cada data lançada.

<u>Entrada</u>	<u>Saída Esperada</u>
Selecionar uma aula em um cronograma	O sistema deve exibir as informações da aula como: sua descrição, duração em minutos, o tipo da aula, horário que ela acontece e seu local.

**TF-07:** Verificar se o sistema é capaz de distribuir as aulas no diário, segundo o cronograma de cada turma, ao longo do semestre.

<u>Entrada</u>	<u>Saída Esperada</u>
----------------	-----------------------



	a definir
--	-----------

**TF-08:** Verificar se o sistema é capaz de permitir o lançamento de aulas extras pelo docente.

<u>Entrada</u>	<u>Entrada</u>
a definir	a definir

**TF-09:** Verificar se o sistema é capaz de registrar as presenças/ausências dos alunos para cada aula no cronograma da turma, clicando em cada célula correspondente. Por padrão, o aluno recebe presença e diante de um clique de mouse, o valor da célula passa a conter um 'F', indicando falta do aluno.

<u>Entrada</u>	<u>Saída Esperada</u>
a definir	a definir

**TF-10:** Verificar se o sistema é capaz de calcular e registrar o total de faltas para cada aluno de uma turma, por solicitação do docente ou ao final de cada mês do semestre letivo.

<u>Entrada</u>	<u>Entrada</u>
a definir	a definir

**TF-11:** Verificar se o sistema é capaz de interoperar com o sistema SAPIENS, ao final de cada mês, para atualizar essa informação.

**TF-12:** Verificar se o sistema é capaz de gerar relatório de faltas acumuladas por turma de cada disciplina ministrada por um docente.

**TF-13:** Verificar se o sistema é capaz de gerar um relatório de faltas acumuladas para todas as turmas de disciplinas nas quais o aluno está matriculado.

**TF-13.1:** Verificar se o sistema é capaz de gerar um relatório resumido, exibindo o total de faltas para cada turma de disciplina na qual o aluno está matriculado.

**TF-13.2** – Verificar se o sistema é capaz de gerar um relatório detalhado, exibindo as datas das faltas (dia/mês/ano) para cada turma de disciplina na qual o aluno está matriculado.

### 3.2.3 Teste da Interface do Usuário

**TI-01:** Navegue através de todas as telas, verificando se cada interface gráfica pode ser rapidamente entendida e facilmente utilizada.

**TI-02:** Verifique que todas as palavras e expressões apresentadas estão em conformidade com as devidas normas sintáticas e gramaticais.

**TI-03:** Verificar se ao clique do mouse na célula referente ao aluno passe a conter o caracter 'F' em caso de falta do mesmo.

### 3.2.4 Teste de Controle de Acesso

**TC-01:** Verificar que usuários não cadastrados não podem acessar informações restritas aos cadastrados.

**TC-02:** Verificar que além do administrador, ninguém mais pode inserir, atualizar ou remover dados do sistema.

**TC-03:** Verificar que os discentes apenas vêem suas próprias informações no sistema.

**TC-04:** Verificar que os docentes vêem suas próprias informações de conta no sistema.

**TC-05:** Verificar que os usuários do sistema podem acessar apenas as funcionalidades e dados associados ao seu próprio tipo de usuário.

### 3.3 Funcionalidades a não serem testadas

Testes de segurança e violação, de stress, de volume, estrutural, de mutantes, dentro outros não serão utilizados nesta etapa do projeto devido ao prazo destinado a entrega e a impossibilidade de integração com os sistemas já existentes no momento.

## 4. Abordagem de teste

Neste plano de teste serão abordadas as ferramentas apresentadas na Tabela 1. Essas ferramentas foram escolhidas de acordo com as necessidades básicas para os testes, sendo focados em apenas em testes funcionais, de banco de dados, controle de acesso e teste de interface de usuário.

**Tabela 1:** Ferramentas de apoio

	<u>Ferramenta</u>
Gerenciamento de Teste	Padrão IEEE 829
Projeto de Teste	JaBUTi
Gerenciamento de Projeto	Google Docs
Ferramentas do SGBD	MySQL Control Center

#### 4.1 Teste de Banco de Dados

Objetivo do Teste:	Garantir que os métodos e processos de acesso ao banco de dados funcionem apropriadamente e sem corrupção dos dados.
Técnica:	<ul style="list-style-type: none"><li>▪ Invocar cada método e processo de acesso ao banco de dados, alimentando cada um com dados ou requisições de dados válidos e inválidos.</li><li>▪ Inspeccionar o banco de dados para garantir que os dados foram populados como pretendido, que todos os eventos do banco de dados ocorreram apropriadamente, ou revisar os dados retornados para garantir que os dados corretos foram recuperados pelas razões corretas.</li></ul>
Critério de Finalização:	Todos os métodos e processos de acesso à base de dados funcionam como projetados e sem nenhuma corrupção de dados.
Considerações Especiais:	<ul style="list-style-type: none"><li>▪ O teste pode necessitar de um ambiente de desenvolvimento ou drivers de SGBD para inserir ou modificar os dados diretamente nas base de dados</li><li>▪ Processos devem ser invocados manualmente</li><li>▪ Bases de dados pequenas ou minimizadas (número de registros limitados) devem ser usados para aumentar a visibilidade de eventos não-aceitáveis.</li></ul>

#### 4.2 Teste Funcional

Objetivo do Teste:	Garantir a funcionalidade apropriada do alvo do teste, incluindo navegação, entrada de dados, processamento, e recuperação.
Técnica:	<p>Executar cada caso de teste, usando dados válidos e inválidos, para verificar o seguinte:</p> <p>Os resultados esperados ocorrem quando dados válidos são usados</p> <p>As mensagens de erro ou aviso apropriadas são exibidas quando dados inválidos são usados.</p> <p>Cada regra de negócio é aplicada apropriadamente</p>

Critério de Finalização:	Todos os testes planejados foram executados. Todos os defeitos identificados foram tratados.
--------------------------	---

#### 4.3 Teste da Interface do Usuário

Objetivo do Teste:	Verificar o que: As avaliações obtidas através do teste deve estar em conformidade com os requisitos funcionais.
Técnica:	Criar ou modificar os testes a fim de verificar os estados dos objeto apropriados para cada tela e objetos da aplicação.
Critério de Finalização:	É verificado que cada tela permaneça consistente com a versão de comparação ou dentro de padrões aceitáveis.

#### 4.4 Teste de Controle de Acesso

Objetivo do Teste:	Verificar que os usuários podem acessar apenas aquelas funções ou dados para os quais o seu tipo de usuário tenha permissão.
Técnica:	Identificar e listar cada tipo de usuário e as funções ou dados para os quais cada tipo tenha permissão.  Criar testes para cada tipo de usuário e verificar cada permissão criando transações específicos para cada tipo de usuário.  Modificar o tipo de usuário e repita os testes para os mesmos usuários. Em cada caso, verificar que funções ou dados adicionais estão corretamente disponíveis ou negados.
Critério de Finalização:	Para cada tipo de usuário conhecido, as funções ou dados apropriados estão disponíveis, e todas as transações funcionam como esperado e rodam nos Testes de Função anteriores.

## 5. Tarefas de teste

## 6. Responsabilidades

<u>Tarefa</u>	<u>Responsáveis</u>
Teste funcional	a definir
Teste de banco de dados	a definir
Teste de interface de usuário	a definir
Teste de controle de acesso	a definir

## 7. Cronograma de testes

<u>Milestone</u>	<u>Data de Início</u>	<u>Data de Término</u>
Planejar teste	10/09/15	11/09/15
Projetar teste	11/09/15	12/09/15
Implementar teste	A definir	-
Executar teste	A definir	-
Avaliar teste	A definir	-

## 8. Aprovações