

**ProgTest: Ambiente de Submissão e Avaliação de  
Trabalhos Práticos**

# Sumário

<b>1</b>	<b>Introdução</b>	<b>2</b>
1.1	Propósito . . . . .	2
1.2	Escopo . . . . .	2
1.3	Definições . . . . .	2
1.4	Visão Geral . . . . .	3
<b>2</b>	<b>Descrição Geral</b>	<b>4</b>
2.1	Perspectiva do Produto . . . . .	4
2.2	Funções do Produto . . . . .	4
<b>3</b>	<b>Requisitos Específicos</b>	<b>5</b>
3.1	Requisitos Funcionais . . . . .	5
3.1.1	Gerais . . . . .	7
3.1.2	Visão Professor . . . . .	8
3.1.3	Visão Aluno . . . . .	10
3.1.4	Submissão . . . . .	10
3.1.5	Avaliação - Detalhar mais . . . . .	10
3.2	Requisitos de <i>Interface</i> Externa . . . . .	11
3.2.1	<i>Interface</i> do Usuário . . . . .	11
3.2.2	<i>Interface</i> de <i>Hardware</i> . . . . .	11
3.2.3	<i>Interface</i> de <i>Software</i> . . . . .	11
3.2.4	<i>Interface</i> de Comunicação . . . . .	11
3.3	Requisitos de Performance . . . . .	11
3.4	Requisitos de Qualidade . . . . .	11
3.4.1	Funcionalidade . . . . .	11
3.4.2	Usabilidade . . . . .	11
3.4.3	Confiabilidade . . . . .	11
3.4.4	Eficiência . . . . .	11
3.4.5	Manutenibilidade . . . . .	11
3.4.6	Portabilidade . . . . .	11

# Capítulo 1

## Introdução

### 1.1 Propósito

Este documento tem como objetivo descrever os requisitos de um ambiente baseado na WEB para submissão e avaliação automática de trabalhos práticos, denominado PROG-TEST. Este documento servirá de base para o projeto, desenvolvimento e manutenção do ambiente.

### 1.2 Escopo

O ambiente fornecerá apoio ao ensino integrado de fundamentos de programação e de teste de software, uma vez que, permitirá a submissão e avaliação dos trabalhos práticos (programas e casos de teste) desenvolvidos pelos alunos.

### 1.3 Definições

- Professor:  
Responsável por uma ministrar uma determinada disciplina e por inserir no ambiente o programa e os casos de teste “oráculos’t’ que serão utilizados na avaliação dos trabalhos.
- Aluno:  
É aquele que irá desenvolver um programa e seus casos de teste e os submeterá ao ambiente.
- Monitor:  
Responsável por auxiliar o professor numa determinada disciplina.
- Trabalhos Práticos: Corresponde ao programa e aos seus respectivos casos de teste que são criados pelo professor e que os alunos devem submeter.
- Ferramentas de teste: Automatiza a aplicação das técnicas e critérios de teste. Deve ser integrado ao ambiente a fim de analisar os trabalhos dos alunos.

## 1.4 Visão Geral

Este documento está organizado da seguinte forma: foi apresentado nessa Seção o propósito e o escopo do sistema, além de definições de alguns termos importantes. Na Seção 2 é apresentada uma descrição geral do ambiente a ser desenvolvido. Na Seção 3 são apresentados os requisitos funcionais, de *interface* externa, de performance e de qualidade.

# Capítulo 2

## Descrição Geral

### 2.1 Perspectiva do Produto

### 2.2 Funções do Produto

O ambiente deve apoiar a submissão e avaliação automática de trabalhos práticos, baseados na atividade de teste desenvolvidos pelos alunos.

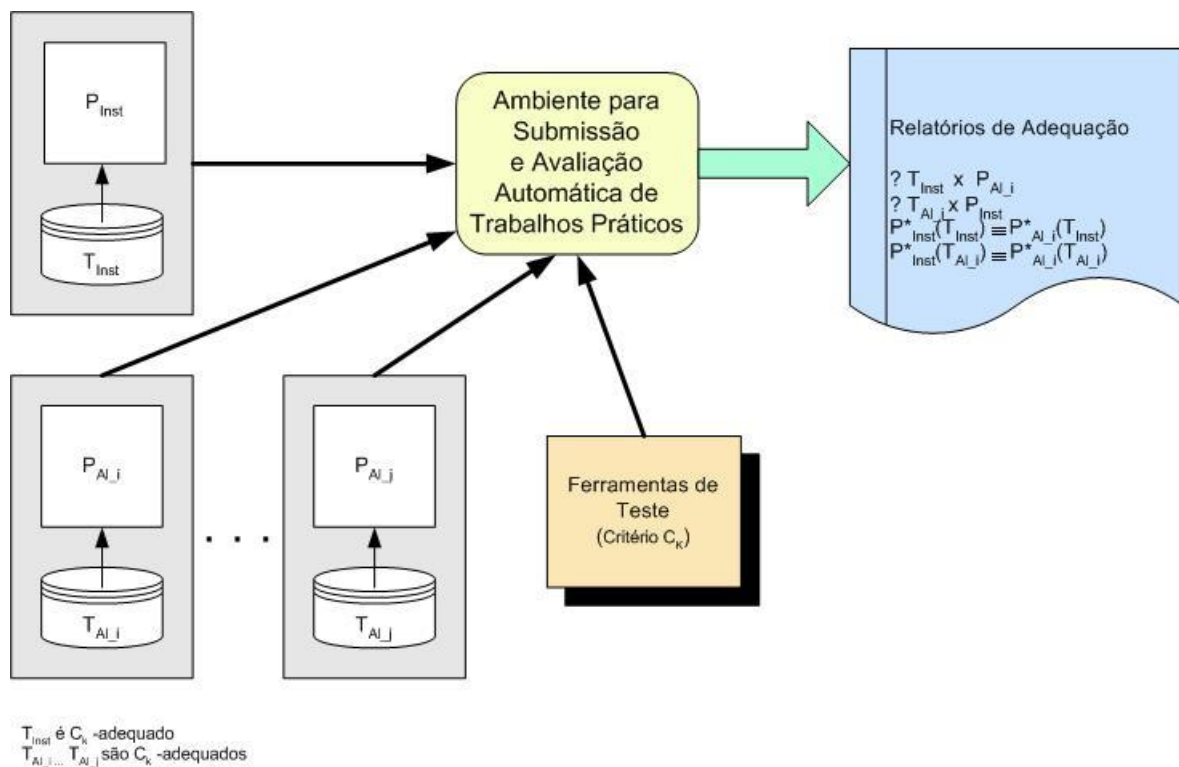


Figura 2.1: Ambiente.

# Capítulo 3

## Requisitos Específicos

### 3.1 Requisitos Funcionais

Os requisitos funcionais do ambiente são:

**Gerais:**

1. Cadastro de usuários (alunos e professores) (inserir, alterar, excluir e visualizar)
  - 1.1 Inserir
  - 1.2 Alterar
  - 1.3 Excluir
  - 1.4 Visualizar
2. Autenticação de usuários
3. Cadastro de disciplinas
  - 3.1 Inserir
  - 3.2 Alterar
  - 3.3 Excluir
  - 3.4 Visualizar

**Professor:**

1. Cadastrar turmas
  - 1.1 Inserir
  - 1.2 Alterar
  - 1.3 Excluir
  - 1.4 Visualizar
2. Adicionar alunos às turmas
3. Cadastrar trabalhos
  - 3.1 Inserir

- 3.2 Alterar
- 3.3 Excluir
- 3.4 Visualizar
- 3.5 Inserir oráculos (programas e casos de teste)
- 4. Visualizar o histórico de cada aluno
  - 4.1 Quais trabalhos foram entregues
  - 4.2 Quais foram os resultados obtidos

**Alunos:**

- 1. Visualizar turma
- 2. Visualizar trabalhos (descrição, datas):
  - 2.1 Realizados (Mostrar resultados da avaliação)
  - 2.2 Em Andamento (Mostrar datas finais, permitir submissão)
- 3. Submeter programas (Permitir submeter qualquer tipo de arquivo ??)
- 4. Submeter casos de teste
- 5. Visualizar resultados da avaliação
  - (a) Por critérios
  - (b) Aceito ou não
  - (c) Nota atribuída ao trabalho

**Submissão:**

- 1. Permitir o upload do arquivo do programa
- 2. Permitir o upload do arquivo com os casos de teste
- 3. Receber somente arquivos do tipo class

**Avaliação:**

- 1. Integrar com a ferramenta de teste Jabuti
- 2. Criar uma fórmula para dar a nota do aluno:
  - 2.1 Baseada nos critérios de teste de software cobertos pela ferramenta Jabuti
  - 2.2 De acordo com o estágio que o aluno está no curso
    - 2.2.1 Pesos devem ser atribuídos a cada critério coberto pela ferramenta Jabuti
    - 2.2.2 Levar em consideração os caminhos não-executáveis???
    - 2.2.3 Mostrar aos alunos a implementação do professor???
    - 2.2.4 Criar um conjunto com todos os casos de teste desenvolvidos por todos os alunos e usar esse conjunto na avaliação dos programas ???

### 3.1.1 Gerais

#### Cadastro de usuários

- Requisito Funcional 1: Inserir usuários
  - Descrição → O sistema deve ser capaz de inserir usuários no sistema.
  - Entrada → Deve serem fornecidos o nome, o número de matrícula, tipo de usuário (professor ou aluno), email, login e senha.
  - Processamento → Verificação se o número de matrícula está correto.
  - Saída → Usuário cadastrado.
- Requisito Funcional 2: Alterar usuários
  - Descrição → O sistema deve ser capaz de alterar usuários no sistema.
- Requisito Funcional 3: Excluir usuários
  - Descrição → O sistema deve ser capaz de excluir usuários do sistema.
- Requisito Funcional 4: Visualizar usuários
  - Descrição → O sistema deve ser capaz de alterar usuários no sistema.
- Requisito Funcional 5: Autenticação de usuários
  - Descrição → O sistema deve ser capaz de fazer a autenticação de usuários.
  - Entrada → Deve serem fornecidos o login e a senha.
  - Processamento → Verificação se o login e a senha estão corretos.
  - Saída → Usuário autenticado no sistema ou uma mensagem de erro indicando que o login ou a senha são inválidos.
- Requisito Funcional 6: Criar disciplinas
  - Descrição → O sistema deve ser capaz de inserir disciplinas no sistema.
  - Entrada → Deve serem fornecidos o nome, o professor responsável, e a quantidade de alunos.
- Requisito Funcional 7: Alterar disciplinas
  - Descrição → O sistema deve ser capaz de alterar disciplinas no sistema.
- Requisito Funcional 8: Excluir disciplinas
  - Descrição → O sistema deve ser capaz de excluir disciplinas do sistema.
- Requisito Funcional 9: Visualizar disciplinas
  - Descrição → O sistema deve ser capaz de visualizar disciplinas no sistema.



### 3.1.2 Visão Professor

- Requisito Funcional 10: Criar turma
  - Descrição → O sistema deve ser capaz de permitir a criação de turmas. Somente o usuário do tipo professor tem o poder de realizar essa função.
  - Entrada → Devem ser fornecidos somente o período em a disciplina é realizada.
- Requisito Funcional 11: Alterar turma
  - Descrição → O sistema deve permitir que os professores alterem os dados das turmas existentes.
  - Entrada → Os dados anteriores da turma.
- Requisito Funcional 12: Excluir turma
  - Descrição → O sistema deve permitir que os professores excluam turmas existentes.
- Requisito Funcional 13: Visualizar turma
  - Descrição → O sistema deve permitir visualização das turmas para todos os usuários.
- Requisito Funcional 14: Adicionar alunos às turmas
  - Descrição → O sistema deve ser capaz de adicionar alunos a uma determinada turma. Somente o usuário tipo professor tem o poder de realizar essa tarefa.
  - Entrada → Devem ser fornecidos o nome e número de matrícula do aluno.
  - Processamento → O sistema deve verificar se o aluno já está cadastrado no sistema. O sistema deve verificar se a quantidade de alunos da turma não foi ultrapassada.
  - Saída → Mensagem indicando sucesso ou mensagem indicando erro.
- Requisito Funcional 15: Inserir trabalhos
  - Descrição → O sistema deve ser capaz de permitir a inserção de trabalhos. Somente o usuário tipo professor tem o poder de realizar essa tarefa.
  - Entrada → Devem ser fornecidos o título e a descrição do trabalho, data de início da submissão, data final da submissão e os arquivos contendo o oráculo(tanto programa e casos de teste).
  - Processamento → O sistema deve verificar se os arquivos são realmente do programa e dos casos de teste fornecidos como oráculo.
  - Saída → Mensagem indicando sucesso ou mensagem indicando erro.
- Requisito Funcional 16: Alterar trabalhos

- Descrição → O sistema deve ser capaz de permitir a alteração de trabalhos existentes. Somente o usuário tipo professor tem o poder de realizar essa tarefa.
  - Entrada → Os dados anteriores do trabalho existente.
- Requisito Funcional 17: Excluir trabalhos
  - Descrição → O sistema deve ser capaz de excluir trabalhos existentes. Somente o usuário tipo professor tem o poder de realizar essa tarefa.
- Requisito Funcional 18: Visualizar trabalhos
  - Descrição → O sistema deve permitir que todos os usuários visualizem os trabalhos existentes.
- Requisito Funcional 19: Inserir alunos
  - Descrição → O sistema deve ser capaz de permitir a inserção de dados dos alunos.
  - Entrada → Devem ser fornecidos o nome, o número de matrícula, login, senha e email.
  - Processamento → Verificação se o número de matrícula está correto.
  - Saída → Usuário cadastrado.
- Requisito Funcional 20: Alterar alunos
  - Descrição → O sistema deve ser capaz de permitir a alteração de dados dos alunos.
  - Entrada → Os dados anteriores do aluno existente.
- Requisito Funcional 21: Excluir alunos
  - Descrição → O sistema deve ser capaz de permitir a exclusão de dados dos alunos.
- Requisito Funcional 22: Guardar histórico dos alunos
  - Descrição → O sistema deve ser capaz de guardar o histórico das submissões e avaliações dos trabalhos realizados de todos os alunos.
- Requisito Funcional 23: Visualizar alunos
  - Descrição → O sistema deve permitir que todos os usuários possam visualizar os dados dos alunos.

### 3.1.3 Visão Aluno

- Requisito Funcional 24: Submeter programas
  - Descrição → O sistema deve permitir que os alunos submetam programas ao sistema.
  - Entrada → O arquivo contendo o programa desenvolvido pelo aluno.
  - Processamento → Deve ser verificado se o arquivo submetido é realmente um arquivo contendo um programa.
  - Saída → Mensagem indicando sucesso ou erro na submissão.
- Requisito Funcional 25: Submeter casos de teste
  - Descrição → O sistema deve permitir que os alunos submetam casos de teste ao sistema.
  - Entrada → O arquivo contendo os casos de teste desenvolvidos pelo aluno.
  - Processamento → Deve ser verificado se o arquivo submetido é realmente um arquivo contendo casos de teste.
  - Saída → Mensagem indicando sucesso ou erro na submissão.
- Requisito Funcional 26: Visualizar resultados da avaliação dos programas e casos de teste
  - Descrição → O sistema deve permitir que todos os usuários visualizem o resultado da avaliação dos trabalhos (por critérios, se foi aceito ou não, nota atribuída) dos alunos.

### 3.1.4 Submissão

- Requisito Funcional 27: Aceitar submissão de programas e casos de teste
  - Descrição → O sistema deve ser capaz de aceitar a submissão de programas e casos de teste.
  - Entrada → Os arquivos desenvolvidos pelos alunos.
  - Processamento → O sistema deve fazer o upload dos arquivos e verificar se os arquivos submetidos estão corretos.
  - Saída → Mensagem indicando sucesso ou erro.

### 3.1.5 Avaliação - Detalhar mais

- Requisito Funcional 28: Avaliar programas e casos de teste
  - Descrição → O sistema deve ser capaz de avaliar os programas e casos de teste submetidos.
  - Entrada → Os arquivos submetidos pelos alunos.
  - Processamento → O sistema deve analisar os arquivos e avaliar os programas e casos de teste de acordo com critérios definidos.
  - Saída → Resultados da avaliação.

## **3.2 Requisitos de *Interface* Externa**

### **3.2.1 *Interface* do Usuário**

### **3.2.2 *Interface* de *Hardware***

### **3.2.3 *Interface* de *Software***

### **3.2.4 *Interface* de Comunicação**

## **3.3 Requisitos de Performance**

## **3.4 Requisitos de Qualidade**

### **3.4.1 Funcionalidade**

### **3.4.2 Usabilidade**

### **3.4.3 Confiabilidade**

O sistema deve ter capacidade para recuperar os dados perdidos da última operação que realizou em caso de falha.

### **3.4.4 Eficiência**

O tempo de processamento de uma operação de avaliação não deve exceder 10 segundos.

### **3.4.5 Manutenibilidade**

O sistema deve ser de fácil manutenção.

### **3.4.6 Portabilidade**

O sistema deve ser portátil.