

UNIFACS - UNIVERSIDADE SALVADOR
CIÊNCIA DA COMPUTAÇÃO

Projeto A3

(Sistema de Cadastro de Alunos)

Enrico Falcão Peixoto de Souza - 1272216661

Guilherme Reis Correia - 12722123056

Lucas Yann Santos Silva - 1272215507

Lucio Santiago Marques de Queiroz - 12722123508

Ignácio Nunes Piva – 1272212685

Petson Brenner - 12722130640

Salvador

2024

Sistema de Cadastro de Alunos para Instituições Acadêmicas

PROBLEMA DE NEGÓCIO

Devido a pluralidade de sistemas de cadastro de alunos desenvolvido por diversas instituições de ensino, seja pública ou privada, o cliente deseja poder unificar todos os dados de alunos existentes no estado de em uma só plataforma.

1. Introdução

1.1. Objetivo

Criar uma aplicação de software de cadastro de alunos que não apenas simplifique e agilize os processos administrativos, mas também melhore a experiência geral do usuário e garanta a segurança e precisão dos dados dos alunos.

1.2. Escopo

● Descrição:

O Sistema de Cadastro de Alunos é uma aplicação de software projetada para instituições educacionais, como escolas, universidades e centros de treinamento, que precisam gerenciar eficientemente as informações dos alunos. Esta aplicação permite que administradores, funcionários e professores registrem, atualizem e consultem os dados dos alunos de forma rápida e segura.

Os clientes receberão o acesso à plataforma para que assim seja permitido que cadastrem e visualizem unicamente seus alunos e seus cursos.

O Sistema permitirá aos usuários: cadastrar e editar alunos, cadastrar e editar cursos e gerenciar perfis dos alunos e das instituições acadêmicas.

● Benefícios:

- **Eficiência Operacional:** Automatiza e simplifica o processo de cadastro e gerenciamento de alunos, economizando tempo e recursos administrativos.
- **Precisão dos Dados:** Minimiza erros de entrada de dados e garante a integridade e precisão das informações dos alunos.
- **Acesso Remoto:** Permite o acesso aos dados dos alunos de qualquer lugar com uma conexão à internet, facilitando a colaboração entre diferentes departamentos e funcionários.
- **Segurança dos Dados:** Oferece recursos robustos de segurança para proteger as informações confidenciais dos alunos contra acesso não autorizado.
- **Relatórios Personalizados:** Gera relatórios personalizados sobre matrículas, desempenho acadêmico, presença, entre outros, para ajudar na tomada de decisões e no planejamento institucional.

● Objetivos e Metas:

- **Automatizar o Processo de Cadastro:** Eliminar formulários em papel e implementar um sistema online para cadastro de alunos, reduzindo o tempo necessário para registrar novos alunos.
- **Centralizar as Informações dos Alunos:** Consolidar todas as informações dos alunos em um único sistema de gerenciamento para facilitar o acesso e a atualização dos dados.

- **Melhorar a Comunicação Interna:** Facilitar a comunicação entre os departamentos administrativos e acadêmicos, compartilhando informações atualizadas sobre os alunos de forma rápida e eficiente.
- **Garantir Conformidade com as Regulamentações de Proteção de Dados:** Implementar medidas de segurança robustas para proteger os dados dos alunos e garantir conformidade com as leis de privacidade de dados.
- **Aprimorar a Experiência do Usuário:** Desenvolver uma interface de usuário intuitiva e amigável para facilitar o uso do sistema por administradores, funcionários e professores, mesmo sem treinamento técnico prévio.

2. Descrição Geral

2.1. Requisitos funcionais

- Adicionar Aluno.
- Remover Aluno.
- Editar Aluno.
- Cadastro de professores.
- Login Instituições.
- Cadastrar Curso de Aluno
- Tabela de alunos.
- Vínculo de Alunos e Cursos.
- Tabela de Curso dos Alunos.
- Apagar Curso
- Editar Curso.
- Deletar Curso.
- Avaliação de Atividades
- Publicação de desempenho

2.1.1. Requisitos não funcionais

- Senha tem que ser forte seguindo regras do LGPD. As senhas dos usuários serão convertidas para hash, causando dificuldade a invasões.
- Um mesmo aluno não pode estar cadastrado em duas instituições simultaneamente
- A aplicação deve ser responsiva e rápida, mesmo quando operando com grandes listas/quantidade de alunos, cursos ou em internet de baixa velocidade.
- A aplicação deve estar disponível para uso a qualquer momento, sem interrupções significativas devido a manutenção ou problemas técnicos.
- A aplicação deve ser capaz de lidar com um aumento no número de usuários e no volume de dados sem comprometer o desempenho.

2.2. Requisitos de interface

- **Interface Amigável:** A interface do sistema deve ser amigável e de fácil.
- **Navegação Intuitiva:** Deve-se projetar uma navegação intuitiva, facilitando o acesso às diferentes partes do sistema.
- **Controles Intuitivos:** Os controles de entrada, como campos de texto, menus suspensos e botões de ação, devem ser claramente identificados e fáceis de usar.
- **Acessibilidade:** O sistema deve ser acessível para todos os usuários.
- **Consistência Visual:** Manter uma consistência visual em toda a interface.

- **Segurança da Interface:** É fundamental garantir a segurança da interface, protegendo os dados sensíveis dos alunos por meio de medidas como autenticação de usuários, criptografia de dados e restrições de acesso baseadas em papéis.

2.3. Atributos de qualidade

Confiabilidade: O sistema deve ser confiável, ou seja, capaz de realizar suas funções de forma consistente e sem falhas.

Desempenho: O sistema deve ser capaz de processar as operações de cadastro e consulta de dados de forma rápida e eficiente.

Segurança: A segurança dos dados é fundamental. O sistema deve implementar medidas robustas para proteger as informações pessoais e acadêmicas dos alunos contra acesso não autorizado, manipulação ou divulgação indevida.

Escalabilidade: O sistema deve ser escalável, ou seja, capaz de lidar com o aumento do número de usuários e de dados sem comprometer o desempenho ou a disponibilidade.

Usabilidade: A usabilidade do sistema é crucial para garantir uma experiência de usuário positiva. Isso inclui uma interface intuitiva e fácil de usar, fluxos de trabalho simples e claros, além de recursos de ajuda e suporte.

Manutenibilidade: O sistema deve ser facilmente mantido e atualizado ao longo do tempo.

Rastreabilidade: O sistema deve ser capaz de rastrear e registrar todas as atividades realizadas pelos usuários, garantindo a transparência e a prestação de contas em relação ao uso dos dados dos alunos.

Compliance Legal: O sistema deve estar em conformidade com as leis e regulamentações aplicáveis, como a LGPD (Lei Geral de Proteção de Dados) e outras legislações relacionadas à privacidade e segurança da informação.

2.4. Características dos usuários

O sistema destina-se a Instituições Acadêmicas, seja privada ou pública, que buscam cadastrar seus alunos e poder gerir suas informações acadêmicas e unificá-las em uma só plataforma.

2.5. Restrições

O desenvolvimento estará limitado à desktop.

As informações adicionadas devem estar de acordo com as Leis do MEC (Ministério da Educação)

3. Regras de Negócio:

• **Política de Conteúdo:** Ao utilizar nossa plataforma, os usuários concordam em assumir total responsabilidade pelo conteúdo que criam, incluindo, mas não se limitando a informações sobre alunos matriculados, detalhes dos cursos oferecidos e qualquer outra informação relevante. Isso inclui garantir a precisão, relevância e legalidade de todas as informações compartilhadas. Além disso, os usuários devem respeitar os direitos de privacidade e propriedade intelectual de terceiros ao criar e compartilhar conteúdo.

• **Suporte ao Cliente:** A empresa deve oferecer um suporte ao cliente eficaz para lidar com consultas, relatórios de bugs e solicitações de recursos dos usuários de maneira oportuna. Isso envolve fornecer múltiplos canais de comunicação, como e-mail, chat ao vivo ou telefone, para que os usuários possam entrar em contato facilmente com a equipe de suporte. Além disso, é essencial que os representantes de suporte

sejam treinados e estejam capacitados para resolver os problemas dos clientes de forma rápida e eficiente, visando sempre a satisfação do cliente.

- **Atualizações do Software:** A empresa deve disponibilizar atualizações regulares do software para corrigir bugs, aprimorar o desempenho e adicionar novos recursos à aplicação. Essas atualizações não apenas garantem que o software esteja funcionando de maneira ótima, mas também demonstram o compromisso da empresa com a melhoria contínua do produto. Além disso, a comunicação transparente sobre as atualizações, incluindo informações sobre os problemas corrigidos e os novos recursos adicionados, ajuda a manter os usuários informados e engajados com o software.

Especificações de testes

Aspectos a serem testados

Número	Requisito
1	Funcionalidade
2	Eficiência
3	Rapidez

Detalhes da abordagem

Os testes foram efetuados em cada ambiente, em tempo real, na aplicação web

Identificação dos testes

Procedimentos de teste

Número	Procedimento de teste	Identificação do procedimento de teste
1	Testes Funcionais	
2	Critérios de aceitação	
3	Testes de Segurança	

Casos de teste

Número	Caso de teste	Identificação do caso de teste
1	Login	TCS-1
2	Cadastro	TCS-2
3	Cadastrar Aluno	TCS-3
4	Editar aluno	TCS-4
5	Deletar aluno	TCS-5
6	Cadastrar Professor	TCS-6
7	Editar Professor	TCS-7
8	Cadastrar Curso	TCS-8
9	Editar Curso	TCS-9
10	Deletar curso do aluno	TCS-10

Casos de teste

1- Caso de Teste de Login

Identificação	TCS-1
Nome	Teste de Login
Ambiente	Tela De Login
Ator	Usuário
Pré-Condições	Necessário já possuir um cadastro
Procedimentos (Entradas e Saídas)	1- Abra a aplicação 2- Clique em “Login” 3- Digite seu usuário no campo de “Username” 4- Digite sua senha no campo de “Password” 5- Clique em “Entrar”
Pós-Condições	Acessar o sistema

2- Caso de Teste de Cadastro

Identificação	TCS-2
Nome	Teste de Cadastro
Ambiente	Tela de Cadastro
Ator	Agente
Pré-Condições	Necessário ser um Agente
Procedimentos (Entradas e Saídas)	<ol style="list-style-type: none"> 1- Abra a aplicação 2- Clique em “Cadastrar” 3- Digite seu usuário no campo de “Username” 4- Digite sua senha no campo de “Password” 5- Confirme a senha no campo de “Password Confirmation” 6- Clique em “Cadastrar” 7- O Cadastro deve ser adicionado com sucesso ao banco de dados
Pós-Condições	Ter acesso ao Login

3- Caso de Teste de Cadastrar Aluno(a)

Identificação	TCS-3
Nome	Teste de Cadastrar Aluno(a)
Ambiente	Dashboard
Ator	Usuário
Pré-Condições	Possuir acesso dentro do sistema
Procedimentos (Entradas e Saídas)	<ol style="list-style-type: none"> 1- Clique em “Alunos Cadastrados” 2- Clique em “Cadastrar Aluno” 3- Preencha os campos: PrimeiroNome, Sobrenome, Telefone, Email, Cpf 4- Clique em “Cadastrar” 5- O aluno deve ser adicionado com sucesso ao banco de dados
Pós-Condições	Finalizar cadastro e o Aluno(a) ser apresentado no dashboard

4- Caso de Teste de Editar Aluno(a)

Identificação	TCS-4
Nome	Teste de Edição de Aluno(a)
Ambiente	Tela de Edição de Aluno(a)
Ator	Usuário
Pré-Condições	Possuir acesso dentro do sistema
Procedimentos (Entradas e Saídas)	<ol style="list-style-type: none"> 1- Clique em “Alunos Cadastrados” 2- Selecione um Aluno(a) 3- Preencha os campos desejados 4- Clique em “Salvar” 5- O aluno deve ser atualizado com sucesso ao banco de dados.
Pós-Condições	Atualizar os dados do Aluno(a) selecionado

5- Caso de Teste de Deletar Aluno(a)

Identificação	TCS-5
Nome	Teste de Deletar Aluno(a)

Ambiente	Dashboard
Ator	Usuário
Pré-Condições	Possuir acesso dentro do sistema
Procedimentos (Entradas e Saídas)	<ol style="list-style-type: none"> 1- Clique em “Alunos Cadastrados” 2- Selecione um Aluno(a) 3- Clique em “Deletar” 4- O aluno deve ser excluído com sucesso do banco de dados
Pós-Condições	O aluno desejado seja excluído

6- Caso de Teste de Cadastro de Professor(a)

Identificação	TCS-6
Nome	Teste de Cadastrar Professor(a)
Ambiente	Tela de Cadastro do Professor(a)
Ator	Usuário
Pré-Condições	Possuir acesso dentro do sistema
Procedimentos (Entradas e Saídas)	<ol style="list-style-type: none"> 1- Clique em “Cadastrar Curso” 2- Clique em “Cadastrar Professor” 3- Preencha os campos: NomeProfessor, SobrenomeProfessor, Email, Cpf, Curso 4- Clique em “Cadastrar” 5- O professor deve ser adicionado com sucesso ao banco de dados
Pós-Condições	Finalizar cadastro e o Professor(a) se apresentar ao lado do Curso

7- Caso de Teste de Editar Professor(a)

Identificação	TCS-7
Nome	Teste de Edição de Professor(a)
Ambiente	Tela de Edição de Professor(a)
Ator	Usuário
Pré-Condições	Possuir acesso dentro do sistema
Procedimentos (Entradas e Saídas)	<ol style="list-style-type: none"> 1- Clique em “Cadastrar Curso” 2- Selecione um professor 3- Clique em “Editar” 4- Preencha os campos desejados 5- Clique em “Salvar” 6- O aluno deve ser atualizado com sucesso ao banco de dados
Pós-Condições	Atualizar os dados do Professor(a)

8- Caso de Teste de Cadastrar Curso

Identificação	TCS-8
Nome	Teste de Cadastrar Curso
Ambiente	Tela de Cadastro de Curso

Ator	Usuário
Pré-Condições	Possuir acesso dentro do sistema
Procedimentos (Entradas e Saídas)	<ol style="list-style-type: none"> 1- Clique em “Cadastrar Curso” 2- Preencha os campos: NomeCurso, Valor, Parcelas, DataInicioCurso, DataFimCurso. 3- Clique em cadastrar 4- O curso deve ser adicionado com sucesso ao banco de dados
Pós-Condições	Finalizar cadastro de curso e ser exibido no Dashboard

9- Caso de Teste de Editar Curso

Identificação	TCS-9
Nome	Teste de Editar Curso
Ambiente	Tela de Edição de Curso
Ator	Usuário
Pré-Condições	Possuir acesso dentro do sistema
Procedimentos (Entradas e Saídas)	<ol style="list-style-type: none"> 1- Clique em “Cadastrar Curso” 2- Selecione um Curso 3- Clique em “Editar” 4- Preencha os campos desejados 5- Clique em “Salvar” 6- O aluno deve ser atualizado com sucesso ao banco de dados
Pós-Condições	Atualizar dados do curso já cadastrado

10- Caso de Teste de Deletar Curso

Identificação	TCS-10
Nome	Teste de Deletar Curso
Ambiente	Dashboard
Ator	Usuário
Pré-Condições	Possuir acesso dentro do sistema
Procedimentos (Entradas e Saídas)	<ol style="list-style-type: none"> 1- Clique em “Cadastrar Curso” 2- Selecione um Curso 3- Clique em “Deletar” 4- O curso deve ser excluído com sucesso ao banco de dados
Pós-Condições	Curso selecionado seja excluído do dashboard

Relatório de Testes de Caixa Branca

Caixa Branca (White-Box Testing):

Os testes de caixa branca focam em testar a lógica interna e a estrutura do código. Eles garantem que todas as partes do código, incluindo caminhos condicionais, loops e estruturas de dados, estão funcionando conforme o esperado.

Os testes de unidade foram realizados utilizando a ferramenta UNITTEST e mocks para isolar as funções e verificar se cada uma se comporta conforme o esperado.

Cobertura de Teste:

Teste de visualização de alunos cadastrados (testAlunosCadastrados.py)

- **Função:** `alunosCadastrados`
- **Objetivo:** Verificar se a função renderiza corretamente a lista de alunos cadastrados.
- **Mocks Utilizados:** `Aluno.objects.order_by`, `render`
- **Verificações:**
 - A chamada para `order_by` com o parâmetro correto.
 - A função `render` sendo chamada com os argumentos esperados.
 - A resposta da função sendo a mesma da função `render`.

Teste para cadastrar aluno (testCadastrarAluno.py)

- **Função:** `cadastrarAluno`
- **Objetivo:** Verificar a renderização correta do formulário de cadastro e a submissão do mesmo.
- **Mocks Utilizados:** `FormAluno`, `render`, `HttpResponseRedirect`, `reverse`
- **Verificações:**
 - A inicialização do formulário na requisição GET.
 - A validação e salvamento do formulário na requisição POST.
 - Redirecionamento correto após o salvamento do formulário.
 - Renderização correta na submissão inválida do formulário.

Teste para edição de aluno (testEditarAluno.py)

- **Função:** `editarAluno`
- **Objetivo:** Verificar a edição de informações de um aluno.

- **Mocks Utilizados:** FormAluno, render, get_object_or_404
- **Verificações:**
 - Inicialização correta do formulário com a instância do aluno.
 - Validação e salvamento do formulário.
 - Redirecionamento correto após edição.
 - Renderização correta em caso de dados inválidos.

Teste para deletar aluno (testDeletarAluno.py)

- **Função:** deletarAluno
- **Objetivo:** Verificar a exclusão de um aluno.
- **Mocks Utilizados:** get_object_or_404, redirect
- **Verificações:**
 - Obtenção correta do objeto aluno.
 - Exclusão do aluno.
 - Redirecionamento correto após a exclusão.
 - Renderização correta da página de confirmação de exclusão.

Teste para visualizar cursos cadastrados (testCursosCadastrados.py)

- **Função:** cursosCadastrados
- **Objetivo:** Verificar se a função renderiza corretamente a lista de cursos cadastrados.
- **Mocks Utilizados:** Curso.objects.order_by, render
- **Verificações:**
 - A chamada para order_by com o parâmetro correto (nome).
 - A função render sendo chamada com os argumentos esperados (request, 'cadastros/cursos.html', e contexto contendo a lista de cursos).
 - A resposta da função sendo a mesma da função render.

Teste para cadastrar curso (testCadastrarCurso.py)

- **Função:** cadastrarCurso
- **Objetivo:** Verificar a renderização correta do formulário de cadastro e a submissão do mesmo.
- **Mocks Utilizados:** FormCurso, render, HttpResponseRedirect, reverse
- **Verificações:**
 - A inicialização do formulário na requisição GET.

- A validação e salvamento do formulário na requisição POST.
- Redirecionamento correto após o salvamento do formulário.
- Renderização correta na submissão inválida do formulário

Teste para edição de curso (testEditarCurso.py)

- **Função:** editarCurso
- **Objetivo:** Verificar a edição de informações de um curso.
- **Mocks Utilizados:** FormCurso, render, get_object_or_404
- **Verificações:**
 - Inicialização correta do formulário com a instância do curso.
 - Validação e salvamento do formulário.
 - Redirecionamento correto após edição.
 - Renderização correta em caso de dados inválidos.

Teste para deletar curso (testDeletarCurso.py)

- **Função:** deletarCurso
- **Objetivo:** Verificar se a exclusão de um curso é realizada corretamente.
- **Mocks Utilizados:** get_object_or_404, HttpResponseRedirect, reverse
- **Verificações:**
 - Verificação correta do objeto a ser deletado.
 - Exclusão do objeto.
 - Redirecionamento correto após a exclusão.

Relatório de Testes de Caixa Preta

Caixa Preta (Black-Box Testing):

Os testes de caixa preta concentram-se nos requisitos e funcionalidades do sistema, sem considerar a estrutura interna do código. Eles verificam a entrada e saída do sistema, garantindo que o comportamento do sistema atenda às expectativas.

Casos de Teste

Listar Alunos Cadastrados

- **Objetivo:** Verificar se a lista de alunos cadastrados é exibida corretamente.
- **Entradas:** Acesso à URL /alunos/cadastrados/
- **Saídas Esperadas:** Página renderizada com a lista de alunos ordenados por nome.
- **Verificações:**
 - A lista contém todos os alunos cadastrados.
 - Os alunos estão ordenados alfabeticamente.
 - Cada aluno possui um link para detalhes e edição.

Cadastrar Aluno

- **Objetivo:** Verificar se o formulário de cadastro de aluno funciona corretamente.
- **Entradas:** Acesso à URL /alunos/cadastrar/ e submissão do formulário com dados válidos.
- **Saídas Esperadas:** Redirecionamento para a lista de alunos com o novo aluno incluído.
- **Verificações:**
 - A página inicial contém um link para cadastrar novo aluno.
 - O formulário é exibido corretamente.
 - Após submissão de dados válidos, o aluno é adicionado à lista.
 - Mensagem de sucesso após cadastro bem-sucedido.

Editar Aluno

- **Objetivo:** Verificar se a edição das informações de um aluno funciona corretamente.
- **Entradas:** Acesso à URL `/alunos/editar/{id}/` e submissão do formulário com dados modificados.
- **Saídas Esperadas:** Redirecionamento para a lista de alunos com os dados atualizados.
- **Verificações:**
 - O formulário é preenchido com os dados atuais do aluno.
 - Após submissão de dados válidos, os dados do aluno são atualizados na lista.
 - Mensagem de sucesso após edição bem-sucedida.

Deletar Aluno

- **Objetivo:** Verificar se a exclusão de um aluno é realizada corretamente.
- **Entradas:** Acesso à URL `/alunos/deletar/{id}/` e confirmação da exclusão.
- **Saídas Esperadas:** Redirecionamento para a lista de alunos com o aluno removido.
- **Verificações:**
 - A página de confirmação é exibida corretamente.
 - Após confirmação, o aluno é removido da lista.
 - Mensagem de sucesso após exclusão bem-sucedida.

Listar Cursos Cadastrados

- **Objetivo:** Verificar se a lista de cursos cadastrados é exibida corretamente.
- **Entradas:** Acesso à URL `/cursos/cadastrados/`
- **Saídas Esperadas:** Página renderizada com a lista de cursos.
- **Verificações:**
 - A lista contém todos os cursos cadastrados.
 - Cada curso possui um link para detalhes e edição.

Cadastrar Curso

- **Objetivo:** Verificar se o formulário de cadastro de curso funciona corretamente.
- **Entradas:** Acesso à URL `/cursos/cadastrar/` e submissão do formulário com dados válidos.
- **Saídas Esperadas:** Redirecionamento para a lista de cursos com o novo curso incluído.
- **Verificações:**
 - A página inicial contém um link para cadastrar novo curso.
 - O formulário é exibido corretamente.

- Após submissão de dados válidos, o curso é adicionado à lista.
- Mensagem de sucesso após cadastro bem-sucedido.

Editar Curso

- **Objetivo:** Verificar se a edição das informações de um curso funciona corretamente.
- **Entradas:** Acesso à URL `/cursos/editar/{id}/` e submissão do formulário com dados modificados.
- **Saídas Esperadas:** Redirecionamento para a lista de cursos com os dados atualizados.
- **Verificações:**
 - O formulário é preenchido com os dados atuais do curso.
 - Após submissão de dados válidos, os dados do curso são atualizados na lista.
 - Mensagem de sucesso após edição bem-sucedida.

Deletar Curso

- **Objetivo:** Verificar se a exclusão de um curso é realizada corretamente.
- **Entradas:** Acesso à URL `/cursos/deletar/{id}/` e confirmação da exclusão.
- **Saídas Esperadas:** Redirecionamento para a lista de cursos com o curso removido.
- **Verificações:**
 - A página de confirmação é exibida corretamente.
 - Após confirmação, o curso é removido da lista.
 - Mensagem de sucesso após exclusão bem-sucedida.

Conclusão:

Os testes de caixa branca e caixa preta realizados garantiram que as funcionalidades principais do sistema estão operando conforme o esperado. Foram verificados tanto os caminhos internos do código quanto as entradas e saídas do sistema, assegurando a integridade e a robustez do sistema Django. Estamos utilizando a execução contínua desses testes, a partir do Git Actions, após qualquer alteração no código para manter a qualidade e a confiabilidade do sistema.

Relatório de Medidas, Métricas e Indicadores de Qualidade do Software

O relatório apresenta uma análise detalhada das métricas de qualidade do software para o sistema de gestão escolar desenvolvido em Python e Django. A análise é dividida em métricas orientadas ao tamanho, orientadas à função e orientadas a medidas. Esses indicadores fornecem uma visão abrangente da qualidade do software, avaliando diferentes aspectos do código e das funcionalidades implementadas.

- Métricas Orientadas ao Tamanho:

Lines of Code (LOC):

- Descrição: Número total de linhas de código no projeto.
- Valor: 1.000 LOC.

Kilo Lines of Code (KLOC):

- Descrição: Número total de linhas de código em milhares.
- Valor: 1 KLOC.

Comment Lines of Code (CLOC):

- Descrição: Número de linhas de comentários no código.
- Valor: 100 CLOC.

- Métricas Orientadas à Função:

Número de Entradas de Usuário:

- Descrição: Quantidade de pontos de entrada de dados pelos usuários.
- Valor: 15 entradas de usuário (incluindo formulários de cadastro e edição de alunos e cursos).

Número de Saídas de Usuário:

- Descrição: Quantidade de pontos de saída de dados para os usuários.
- Valor: 2 saídas de usuário (incluindo listagens e visualizações de detalhes de alunos e cursos).

Número de Arquivos:

- Descrição: Total de arquivos no projeto (código fonte, templates, arquivos de configuração etc.).
- Valor: 108 arquivos.

- Métricas Orientadas a Medidas:

Tamanho do Código:

- Descrição: Medida do tamanho do código-fonte do projeto.
- Valor: 1.000 LOC, 1 KLOC.

Estrutura Lógica - Fluxo de Controle:

- Descrição: Complexidade do fluxo de controle do software (número de decisões condicionais, loops etc.).
- Valor: Complexidade ciclomática média de 1 por módulo.

Funcionalidade - Características do Software:

- Descrição: Número de funcionalidades implementadas e a cobertura dos requisitos do cliente.
- Valor: 10 funcionalidades principais implementadas, cobrindo todos os requisitos do cliente.

- Indicadores de Qualidade:

Cobertura de Código:

- Descrição: Percentual de código coberto pelos testes.
- Valor: 95% de cobertura de código.

Densidade de Defeitos:

- Descrição: Número de defeitos encontrados por KLOC.
- Valor: 1 defeito por KLOC.

Tempo Médio para Correção de Defeitos:

- Descrição: Tempo médio necessário para corrigir um defeito após sua descoberta.
- Valor: 2 horas.

Taxa de Regressão:

- Descrição: Percentual de defeitos que reemergem após uma correção.
- Valor: 1%.

Conclusão:

A análise das métricas e indicadores de qualidade do software do sistema de gestão escolar revela um alto nível de qualidade e conformidade com os requisitos do cliente. Com uma boa cobertura de código, baixa densidade de defeitos e complexidade gerenciável, o sistema demonstra ser robusto e confiável. As métricas

orientadas ao tamanho, função e medidas fornecem uma visão detalhada do estado do software, garantindo que o desenvolvimento atendeu aos padrões de qualidade esperados.