

Plano de testes

Nome do sistema:

Sistema de Controle de Escola de música

Resumo do sistema:

O projeto de software para apoio ao *Espaço Cultural Tocando em Você* conta com quatro módulos: um que permite o cadastro e listagem de funcionários, alunos e professores que participam da escola; o segundo para realizar o controle de pagamentos das mensalidades; um terceiro que é responsável pela marcação dos horários das aulas e o último que permite acompanhar a situação financeira.

A página de cadastro deve permitir a inclusão de um novo participante (aluno, professor ou funcionário) somente quando todas as informações obrigatórias forem preenchidas e valida o preenchimento correto dessas informações.

A página de controle de pagamentos salva os valores mensais efetuados por cada aluno, por curso realizado, podendo incluir observações relacionadas a quitação dos valores.

A página de marcação dos horários é realizada de acordo com a disponibilidade dos professores + sala de aula ao longo da semana.

Por fim, a página de controle financeiro gera relatórios da situação financeira dos alunos em um determinado período, para um dado curso.

Objetivos do plano de teste:

Esse documento do plano de testes do Sistema de Controle de Escola de Música é composto dos seguintes objetivos:

- Identificar os tipos de dados que devem ser aceitos pelo cadastro no banco de dados (*armazenados em arquivos JSON*).
- Recomendar estratégias de teste que funcionam melhor no sistema.

- Listar a forma de apresentação dos resultados dos testes.
- Listar a equipe de pessoas responsável pelos testes.
- Criar o cronograma de testes.

Requisitos de teste (casos de teste):

Nesta etapa, é descrito abaixo o que será verificado em cada um dos testes utilizados no sistema:

1. Teste de banco de dados *(armazenados em arquivos JSON)*

- Verifica se ocorre a conexão com o banco de dados.
- Verifica se é possível cadastrar novos participantes (alunos, professores ou funcionários) no banco de dados.
- Verifica se é possível obter os dados desses participantes cadastrados.
- Verifica se é possível cadastrar no banco de dados os pagamentos das mensalidades dos alunos em cada curso que participa.
- Verifica se é possível realizar a marcação dos horários semanais das aulas, no banco de dados.
- Verifica se é possível consultar no banco de dados a situação financeira dos alunos em um determinado período, para um dado curso.

2. Teste funcional

- Verifica se o programa inicia e finaliza corretamente sempre que o usuário desejar.
- Verifica se o sistema permite o cadastro de um novo participante (aluno, professor ou funcionário) com as seguintes informações obrigatórias: nome, CPF, identidade, UF da identidade, data de nascimento e endereço.
- Verifica se o sistema valida se os campos obrigatórios foram preenchidos corretamente antes de permitir o cadastro do participante.
- Verifica se as mensagens de preenchimento inválido das informações digitadas no cadastro de participantes aparecem quando deveriam.

- Verifica se o sistema permite a visualização dos dados dos participantes cadastrados no sistema.
- Verifica se o sistema preenche automaticamente na página os dados dos participantes cadastrados no sistema (listagem).
- Verifica se o sistema permite a seleção de um aluno, curso, mês e ano cadastrados no banco de dados para associar ao pagamento de uma mensalidade.
- Verifica se o sistema permite o cadastro de um novo pagamento com as seguintes informações obrigatórias: aluno, curso, mês, ano e valor.
- Verifica se o sistema valida se os campos obrigatórios foram preenchidos corretamente antes de permitir o registro do pagamento de uma mensalidade.
- Verifica se as mensagens de preenchimento inválido das informações digitadas no registro do pagamento de uma mensalidade aparecem quando deveriam.
- Verifica se o sistema permite a seleção de um aluno, curso, professor, sala, dia da semana e horário cadastrados no banco de dados para associar a marcação de uma aula.
- Verifica se o sistema permite o cadastro de uma nova aula marcada com as seguintes informações obrigatórias: aluno, curso, professor, sala, dia da semana e horário.
- Verifica se o sistema valida se os campos obrigatórios foram preenchidos corretamente antes de permitir o registro da nova aula marcada.
- Verifica se as mensagens de preenchimento inválido das informações digitadas no registro da aula marcada aparecem quando deveriam.
- Verifica se o sistema permite a seleção de um aluno e curso para associar a consulta da situação financeira.
- Verifica se o sistema permite a pesquisa da situação financeira com as seguintes informações obrigatórias: data de início, data de fim, aluno e curso.
- Verifica se o sistema valida se os campos obrigatórios foram preenchidos corretamente antes de permitir a pesquisa da situação financeira.

- Verifica se as mensagens de preenchimento inválido das informações digitadas na pesquisa da situação financeira aparecem quando deveriam.

3. Teste de interface do usuário

- Navega por todas as funcionalidades do programa, testando seu funcionamento.
- Verifica se a interface é de fácil utilização e aprendido.
- Verifica se a interface mantém alinhamentos e proporções estipulados no protótipo.
- Verifica se todo o texto apresentado na interface está conforme as normas de grafia e as regras gramaticais.

4. Teste de carga

- Verifica o sistema com 1 usuário realizando cadastro/consulta na base de dados.
- Verifica o sistema com 10 usuários realizando cadastro/consulta na base de dados.
- Verifica o sistema com 20 usuários realizando cadastro/consulta na base de dados.
- Verifica o sistema com 50 usuários realizando cadastro/consulta na base de dados.

5. Teste de segurança

- *Só será possível realizar o mesmo quando for implementado o módulo de controle de acesso ao sistema.*

Estratégias de teste:

1. Teste de banco de dados

Objetivo do teste	Garantir que os dados cadastrados estejam sendo salvos corretamente no banco de dados.
Técnica	Invocar cada método de cadastro ao banco de dados com valores válidos e inválidos.
Critério de finalização	Todos os acessos à base de dados funcionam como projetados e sem perda de informações.
Considerações especiais	Processos devem ser invocados manualmente.

2. Teste funcional

Objetivo do teste	Garantir que as principais funções do programa estejam funcionando.
Técnica	<p>Executar todos os casos de uso, usando dados válidos e inválidos, para verificar se:</p> <ul style="list-style-type: none">• Cada regra de negócio é aplicada corretamente.• As mensagens de erro aparecem sempre que necessário e da forma como foram projetadas.• Quando dados válidos são colocados, ocorrem os resultados esperados.
Critério de finalização	<p>Todos os testes foram aplicados com êxito.</p> <p>Todos os erros identificados foram corrigidos ou reportados.</p>
Considerações especiais	Aqui, pode ser reportado algo que ajude na avaliação final dos testes funcionais.

3. Teste de interface do usuário

Objetivo do teste	Garantir que a interface apresente o mesmo formato projetado no protótipo e respeite as regras de usabilidade definidas pela equipe de desenvolvimento e seus gerentes.
Técnica	Testar todos os elementos de cada janela da interface, especialmente botões, textos e mensagens de erros. Modificar o tamanho das janelas para ver se funcionam nos dispositivos para os quais foram projetadas.
Critério de finalização	As janelas funcionam sem erros em todos os dispositivos para os quais foram projetadas. Botões e mensagens de erro, ou avisos, abrem corretamente sempre que for necessário. A interface respeita as regras de usabilidade do projeto do sistema.
Considerações especiais	Nenhuma

4. Teste de carga

Objetivo do teste	Testar o tempo de resposta do sistema, alterando o número de usuários acessando ao mesmo tempo.
Técnica	Testar o tempo de resposta do sistema, alterando o número de usuários acessando ao mesmo tempo.
Critério de finalização	O sistema não apresentar falhas nem perda de acesso com até 50 usuários ao mesmo tempo.
Considerações especiais	Deve-se ter toda a base de dados em uma mesma máquina recebendo todos os acessos para que o teste tenha validade.

5. Teste de segurança

Só será possível realizar o mesmo quando for implementado o módulo de controle de acesso ao sistema.

Equipe:

Cargo	Responsabilidades
Gerente	<ul style="list-style-type: none">• Define a equipe de teste• Indica direcionamento de funções• Fornece relatórios• Adquire recursos necessários
<i>Test designer</i>	<ul style="list-style-type: none">• Identifica, prioriza e implementa os casos de teste• Gera o plano de teste• Cria o modelo de cada teste
<i>Designer gráfico/visual</i>	<ul style="list-style-type: none">• Responsável pela elaboração dos testes de interface
Testador	<ul style="list-style-type: none">• Executa os testes• Registra os resultados• Documenta e solicita mudanças
Gerente de banco de dados	<ul style="list-style-type: none">• Prepara o banco de dados para os testes• Administra os dados durante os testes
Implementador	<ul style="list-style-type: none">• Implementa as classes de teste

Infraestrutura de teste:

Recursos de infraestrutura
Servidor de banco de dados MySQL
Apache NetBeans IDE 23
Terminais de teste Cinco PCs (computadores pessoais) – não é necessária conexão à Internet

Cronograma:

Etapas	Data de início	Data de término
Planejar teste	16/04/2025	18/04/2025
Teste estrutural	19/04/2025	21/04/2025
Teste de banco de dados	22/04/2025	22/04/2025
Teste funcional	23/04/2025	23/04/2025
Teste de interface	24/04/2025	24/04/2025
Teste de carga	25/04/2025	25/04/2025
Teste de segurança	26/04/2025	26/04/2025
Avaliar teste	27/04/2025	28/04/2025