

# Gerenciamento de frequência e autoavaliação

Antônio Cauê O. Morais

*Departamento*  
*UFERSA*

Pau dos Ferros, Brasil

antonio.morais36234@alunos.ufersa.edu.br

Karlos Wiliam R. Marques

*Departamento*  
*UFERSA*

Pau dos ferros, Brasil

karlos.marques@alunos.ufersa.edu.br

Davi S. Soares

*Departamento*  
*UFERSA*

Pau dos Ferros, Brasil

davi.soares@alunos.ufersa.edu.br

**Resumo**—Neste artigo exploraremos o processo de elaboração de um sistema de gerenciamento de frequência e autoavaliação. Descrevendo os recursos e tecnologias usados, bem como os requisitos e testes implementados para garantir a qualidade do sistema.

**Index Terms**—Gerenciamento de Frequência, Autoavaliação, Teste de Software, Requisitos do sistema.

## I. INTRODUÇÃO

A frequência em sala de aula é um tema que quando entra em discussão, há quem defenda argumentando sua importância para o desempenho acadêmico e quem discute sua efetividade. O ingresso no ensino acadêmico é um processo que enfrenta diversos obstáculos, como a evasão, que ocorre quando o aluno deixa de frequentar as aulas durante o período letivo, a má gerência dos estudos, a falta de disciplina e foco, são problemas que podem impactar negativamente o aprendizado e a vida acadêmicas do aluno. No estudo de Silva Filho et al. (2007) sobre a evasão no ensino superior brasileiro, os autores observaram:

A evasão estudantil no ensino superior é um problema complexo que afeta não apenas o resultado dos sistemas educacionais, mas também tem implicações sociais, acadêmicas e econômicas significativas. As perdas de estudantes que iniciam seus cursos, mas não conseguem concluí-los, representam desperdícios de recursos públicos no setor público e uma importante perda de receitas no setor privado. Além disso, a evasão gera ociosidade de professores, funcionários, equipamentos e espaço físico, impactando negativamente a eficiência e a qualidade do ensino superior no Brasil (??).

Como proposta para minimizar esse problema e promover uma abordagem mais satisfatória na gestão da frequência e no desempenho dos estudantes do ensino superior, neste artigo, trataremos de um sistema de gerenciamento de faltas e autoavaliação com o objetivo de auxiliar os estudantes a terem um melhor controle de suas frequências, com um instrumento que permita a análise eficaz do seu comportamento em sala de aula. Para que, com base nesses dados, o aluno possa escolher a melhor maneira de melhorar seu rendimento acadêmico, tanto na frequência como no rendimento das disciplinas que está cursando.

## II. DESENVOLVIMENTO

A disciplina se mostra um fator essencial no ambiente acadêmico, a aquisição de conhecimento se dá a partir de uma série de fatores, dentre eles, a constância. Ao ingressarmos no ensino superior nos deparamos com problemáticas das quais são impedimentos para a continuação e manutenção da disciplina, adversidades da vida adulta acabam por impedir participação frequente em sala de aula. Com o objetivo de minar esse problema, procuramos desenvolver um sistema que dê ao usuário um panorama de suas faltas nas disciplinas que está cursando, assim como a possibilidade de analisar relatórios de seu desempenho nas disciplinas, tais relatórios serão baseados na autoavaliação que poderá ser realizada pelo sistema.

O gerenciamento de frequência em sala de aula é uma proposta para ajudar o aluno a possuir um maior controle sobre suas faltas, já que, em muitos casos, as faltas são uma das causas de reprovação, com ele o usuário conhecerá o limite de faltas que pode possuir de acordo com as regras do curso e será alertado caso esteja próximo desse limite.

A autoavaliação é um recurso que auxilia na reflexão e na avaliação pessoal do desempenho do aluno na disciplina. Na pesquisa de Nicol e Macfarlane-Dick (2006), os autores discutem a importância da autoavaliação no processo de aprendizagem:

Autoavaliação é uma parte vital do ciclo de avaliação formativa. Ela envolve os alunos a refletir sobre seus próprios desempenhos e aprender a partir dessa reflexão, o que contribui para o desenvolvimento da metacognição. A autoavaliação pode ocorrer em muitas formas, desde a avaliação de metas de desempenho pessoal até a autoavaliação das estratégias de aprendizado e, como tal, é um componente-chave da autoeficácia, ou seja, a crença do aluno em sua própria capacidade de realizar uma tarefa (p. 199).

Neste trecho, os autores destacam a importância da autoavaliação como parte integrante da avaliação formativa e como um meio pelo qual os alunos podem melhorar seu desempenho e desenvolver a autoeficácia. Eles também mencionam que a autoavaliação pode ocorrer de várias formas, enfatizando sua versatilidade como uma ferramenta de aprendizado.

Na fase de desenvolvimento do projeto se deu inicialmente pela criação das classes previamente definidas no diagrama. Cada classe do programa é responsável por uma parte da funcionalidade do código, foram criadas 9 classes que compõem todo o programa, dessas classes, Menu, Frequência, Disciplina e Relatório Geral, são as que fundamentam as principais funcionalidades do código.

Todas as informações do funcionamento do software, ou seja, as informações de cada disciplina, suas presenças ou faltas nas aulas e sua autoavaliação durante as mesmas ficam concentradas em um banco de dados montado para suprir essas necessidades. Ele armazena esses dados para que todas as funcionalidades do software possam acessá-los. Todos os dados informados pelo usuário são armazenados em um banco de dados criado com tabelas responsáveis por cada parte do projeto, com as seguintes nomenclaturas:

- Tabela **Disciplina** (com **codigo** como chave primária, **nome** e **cargaHoraria** sendo os atributos dessa tabela)
- Tabela **Frequencia** (com **frequencia\_id** como chave primária, **data**, **presencaAusencia**, além da chave estrangeira **codigo** da tabela **Disciplina** como atributos dessa tabela)
- Tabela **RelatorioGeral** (com **relatorio\_id** como chave primária, **porcentagemDeFrequencia**, além da chave estrangeira **codigo** da tabela **Disciplina** sendo os atributos dessa tabela)
- Tabela **Autoavaliacao** (com **autoavaliacao\_id** como chave primária, **comentario**, além da chave estrangeira **frequencia\_id** da tabela **Frequencia** sendo os atributos dessa tabela).

A classe App.java, onde o método main está localizado é a responsável por chamar a classe Menu.java que é responsável por exibir o menu opções e aplicar os métodos que acessam as demais funcionalidades do sistema, como é o caso das opções de cadastro das disciplinas, sendo responsável por solicitar do usuário os dados de cada matéria, dados esses que são: código, nome e carga horária de cada disciplina. O usuário pode no primeiro uso realizar o cadastro de todos os seus componentes curriculares no primeiro uso, mas se necessário, é possível voltar em outro acesso e cadastrar um novo componente.

O método DisciplinaChooser da classe Menu.java é o principal mecanismo desta classe, nele, se concentra a lógica de gestão das disciplinas, como criação, seleção e exclusão, dependendo da entrada do usuário e do estado atual do sistema em relação às disciplinas registradas e a chamada dos outros métodos da classe responsável por tudo esse gerenciamento como createDisciplina() e deleteDisciplina(). Além desse método, a funcionalidade de registro e avaliação de frequência também está na classe Menu.java.

A classe Frequencia.java contém métodos para acessar, armazenar e manipular as tuplas do banco de dados, sendo responsável por armazenar as informações sobre as presenças na sala de aula. A importância dessa classe se dá, pois acompanha o progresso dos alunos, sendo crítico para avaliar o aluno. Com Frequencia.java conseguimos obter um registro preciso de quando o aluno está presente ou não. Os métodos contidos

nesta classe desempenham funções, como a impressão de novos registros de frequência, a criação de informações sobre a frequência passada e a atualização das informações conforme necessário. Além disso, a classe Frequencia.java serve como uma interface entre o banco de dados e outras partes do sistema que precisam acessar esses dados.

A classe Disciplina.java contém métodos, nas quais no programa permitem gerenciar informações relacionadas a disciplinas em um sistema acadêmico, incluindo a criação, exclusão, consulta e impressão de detalhes das disciplinas. Além disso, eles demonstram a interação com um banco de dados para armazenar e recuperar essas informações.

A classe Autoavaliacao.java é a responsável por conter os métodos que acessam, criam, excluem e verificam as autoavaliações no banco de dados, com o objetivo de permitir o aluno de expressar suas opiniões e anseios sobre o seu desempenho acadêmico,

### III. FUNCIONALIDADES DO SISTEMA

Nesta seção, descrevemos as principais funcionalidades do sistema que foram desenvolvidas para atender às necessidades dos estudantes, educadores e instituições de ensino. Abaixo, apresentamos uma lista dos requisitos do sistema que nortearam o desenvolvimento:

- **Registro de presenças e faltas:** Permitir o registro fácil e rápido de presenças e faltas dos alunos em cada aula de todas as disciplinas.
- **Cálculo do limite de faltas:** Calcular automaticamente o limite de faltas permitidas em cada matéria, com base nas regras estabelecidas pela instituição de ensino.
- **Notificação de proximidade de reprovação:** Alertar o aluno quando estiver se aproximando do limite de faltas para reprovação.
- **Histórico de frequência:** Armazenar o histórico de frequência de cada aluno em todas as disciplinas, permitindo a visualização de padrões de presença e ausência ao longo do tempo.
- **Autoavaliação personalizada:** Fornecer um espaço para que os alunos façam suas autoavaliações, registrando seus sentimentos, emoções e pensamentos durante as aulas.
- **Relatórios e análises:** Gerar relatórios e análises com base nas informações de frequência e autoavaliação dos alunos.
- **Acesso fácil e intuitivo:** A aplicação deve ter uma interface intuitiva para que os alunos realizem as tarefas de gerenciamento de frequência e autoavaliação de forma simples.
- **Segurança e privacidade:** Garantir a segurança dos dados dos alunos e respeitar a privacidade das informações fornecidas nas autoavaliações.
- **Acesso multiplataforma:** A aplicação deve ser acessível em diferentes dispositivos (computadores, smartphones e tablets).
- **Desempenho:** A aplicação deve ser responsiva e ter um desempenho satisfatório, mesmo com um grande número de usuários.

- **Confiabilidade:** Garantir que o sistema funcione de forma confiável e esteja disponível para os usuários sempre que necessário.
- **Integração com sistemas acadêmicos:** Se possível, a aplicação deve ser integrada com outros sistemas acadêmicos existentes na instituição.
- **Escalabilidade:** O sistema deve ser capaz de lidar com um número crescente de usuários e dados sem comprometer o desempenho.

#### IV. METODOLOGIA

##### Definição e Justificativa dos Testes

**Definição dos Testes:** Neste projeto de gerenciamento de faltas e autoavaliação, planejamos realizar uma série de testes para garantir que o sistema atenda aos requisitos e funcione conforme o esperado. Os principais tipos de testes incluirão:

- **Testes de Unidade:** Esses testes se concentram na verificação individual de componentes específicos do sistema, como o módulo de registro de frequência e o sistema de autoavaliação. Eles ajudarão a identificar erros em nível de código e garantirão a funcionalidade adequada das partes isoladas do sistema.
- **Testes de Integração:** Os testes de integração serão conduzidos para garantir que todos os módulos e funcionalidades se integrem de maneira harmoniosa. Isso incluirá a verificação de comunicações entre componentes e a interação eficaz entre os módulos do sistema.
- **Testes de Desempenho:** Os testes de desempenho visam avaliar a capacidade do sistema de lidar com a carga de usuários esperada. Isso garantirá que o sistema mantenha um desempenho adequado mesmo durante períodos de alta demanda, como o início do semestre.

**Justificativa dos Testes:** Os testes são fundamentais para o sucesso deste projeto de gerenciamento de faltas e autoavaliação por várias razões:

- **Garantia de Qualidade:** Os testes ajudarão a identificar e corrigir erros, garantindo que o sistema seja confiável e eficiente.
- **Conformidade com Requisitos:** Os testes de aceitação garantirão que o sistema atenda aos requisitos dos usuários, garantindo sua utilidade e eficácia.
- **Desempenho Adequado:** Os testes de desempenho assegurarão que o sistema funcione sem problemas, mesmo sob carga máxima, proporcionando uma experiência consistente aos usuários.
- **Segurança e Privacidade:** Os testes de segurança garantirão que os dados dos estudantes sejam protegidos contra ameaças e violações de segurança.
- **Melhoria Contínua:** Os testes irão fornecer feedback valioso para aprimoramentos futuros do sistema, garantindo que ele permaneça atualizado e eficaz ao longo do tempo.

A realização desses testes é crucial para a qualidade e confiabilidade do nosso sistema de gerenciamento de faltas e autoavaliação, garantindo que ele atenda às expectativas dos usuários e ofereça uma solução sólida para os desafios educacionais enfrentados.

#### V. TESTES REALIZADOS

*A. Título: Teste de unidade do método responsável pelo registro de presenças e faltas nas disciplinas*

**Introdução:** O teste de unidade desempenha um papel crucial no desenvolvimento de software, garantindo que cada componente individual funcione corretamente. Neste caso específico, o teste visa verificar a precisão do registro de presenças e faltas em disciplinas. Isso é vital para assegurar a integridade dos dados e o bom funcionamento do sistema em questão.

**Descrição do Teste:** O teste se concentra na funcionalidade de registro de presenças e faltas em disciplinas, que é um aspecto fundamental do sistema de gerenciamento acadêmico. Ele simula a criação de uma nova disciplina de teste e registra uma frequência específica para essa disciplina. O cenário de teste inclui a verificação de valores específicos relacionados à data, presença/ausência e contagem de faltas.

##### Passos do Teste:

- 1) Estabelecer uma conexão com o banco de dados.
- 2) Criar uma nova disciplina de teste no sistema.
- 3) Registrar uma frequência para a disciplina em uma data específica, indicando uma falta completa.
- 4) Verificar se os valores registrados correspondem aos valores esperados.

**Resultados Esperados:** Espera-se que o teste valide a precisão do registro de presenças e faltas. Isso inclui a verificação de valores específicos, como a data correta, o status de presença/ausência e o número de faltas registradas.

**CrITÉRIOS de Sucesso e Falha:** Sucesso: Os valores registrados correspondem exatamente aos valores esperados, indicando que o sistema registra adequadamente as presenças e faltas. Falha: Qualquer divergência nos valores registrados em relação aos valores esperados indica um possível problema no registro de presenças e faltas.

**Observações e Considerações:** É fundamental garantir que a conexão com o banco de dados esteja funcionando corretamente, para que os dados possam ser armazenados e recuperados durante o teste. Além disso, todos os passos devem ser executados em um ambiente controlado para garantir resultados precisos.

**Conclusão:** O teste de unidade para o registro de presenças e faltas em disciplinas é uma etapa crítica para garantir a funcionalidade correta do sistema de gerenciamento acadêmico. Ao verificar a precisão do registro, podemos garantir a integridade dos dados e a confiabilidade das informações mantidas pelo sistema. Esse teste de unidade desempenha um papel fundamental na garantia da qualidade do produto final e na melhoria da experiência do usuário.

*B. Título: Teste de Unidade para Autoavaliação de Frequência em Disciplinas*

**Introdução:** O teste de unidade desempenha um papel crucial no processo de desenvolvimento de software, ajudando a garantir que os componentes individuais funcionem conforme o esperado. Neste contexto, o teste de unidade em questão

tem como objetivo verificar a funcionalidade de autoavaliação de frequência em disciplinas. Isso é essencial para garantir a precisão das autoavaliações e a integridade dos dados dentro do sistema de gerenciamento acadêmico.

**Descrição do Teste:** O teste concentra-se na funcionalidade de atribuir uma autoavaliação específica a uma frequência de disciplina. Ele simula a criação de uma nova disciplina de teste, o registro de uma frequência específica e a atribuição de uma autoavaliação a essa frequência. O teste verifica se os valores da autoavaliação correspondem aos valores esperados, garantindo a precisão do processo de autoavaliação no sistema.

#### **Passos do Teste:**

- 1) Estabelecer uma conexão com o banco de dados.
- 2) Criar uma nova disciplina de teste no sistema.
- 3) Registrar uma frequência para a disciplina em uma data específica, indicando uma falta completa.
- 4) Atribuir uma autoavaliação a essa frequência.
- 5) Verificar se os valores da autoavaliação correspondem aos valores esperados.

**Resultados Esperados:** Espera-se que o teste valide a precisão do processo de autoavaliação. Isso inclui a verificação de valores específicos, como o comentário da autoavaliação e a correspondência do ID da frequência associada.

**Critérios de Sucesso e Falha:** Sucesso: Os valores da autoavaliação correspondem exatamente aos valores esperados, indicando que o sistema registra corretamente as autoavaliações associadas a uma determinada frequência. Falha: Qualquer divergência nos valores da autoavaliação em relação aos valores esperados indica um possível problema no processo de autoavaliação ou no registro de dados.

**Observações e Considerações:** É crucial garantir que a conexão com o banco de dados esteja funcionando corretamente para que os dados possam ser armazenados e recuperados durante o teste. Além disso, a consistência dos dados de entrada e o funcionamento adequado das consultas SQL são de extrema importância para garantir a precisão dos resultados do teste.

**Conclusão:** O teste de unidade para a autoavaliação de frequência em disciplinas é essencial para garantir a funcionalidade correta do sistema de gerenciamento acadêmico. Ao verificar a precisão do processo de autoavaliação, podemos garantir a integridade dos dados e a confiabilidade das informações mantidas pelo sistema. Esse teste de unidade desempenha um papel fundamental na garantia da qualidade do produto final e na melhoria da experiência do usuário.

### **C. Título: Teste de Unidade para Entrada de Float do Usuário**

#### **Introdução:**

O teste de unidade desempenha um papel crucial na garantia da funcionalidade correta de métodos que lidam com a entrada do usuário. Neste caso, o teste tem como objetivo verificar a precisão da captura de um valor *float* fornecido pelo usuário por meio do método *getUserFloat* em um sistema específico.

#### **Descrição do Teste:**

O teste concentra-se na captura de um valor *float* do usuário por meio de uma simulação de entrada de dados.

O cenário envolve a simulação de um valor *float* de entrada específico e a verificação se o valor capturado corresponde ao valor esperado, levando em consideração limites superiores e inferiores predefinidos.

#### **Passos do Teste:**

- 1) Simular a entrada de um valor *float* específico.
- 2) Capturar o valor *float* inserido pelo usuário por meio do método *getUserFloat*.
- 3) Verificar se o valor capturado corresponde ao valor esperado dentro dos limites definidos.

#### **Resultados Esperados:**

Espera-se que o teste valide a precisão da captura de valores *float* fornecidos pelo usuário, garantindo que o sistema consiga processar corretamente os valores de ponto flutuante inseridos.

#### **Critérios de Sucesso e Falha:**

**Sucesso:** O valor *float* capturado corresponde exatamente ao valor esperado dentro dos limites definidos.

**Falha:** Qualquer divergência entre o valor capturado e o valor esperado, ou se o valor estiver fora dos limites predefinidos, indica um possível problema na captura de valores *float* do usuário.

#### **Observações e Considerações:**

É fundamental garantir que a simulação de entrada de dados seja realizada corretamente para fornecer o valor *float* desejado ao método de teste. Além disso, os limites definidos devem ser cuidadosamente verificados para garantir que o método *getUserFloat* restrinja corretamente os valores de entrada conforme necessário.

#### **Conclusão:**

O teste de unidade para a captura de valores *float* do usuário desempenha um papel crucial na validação da funcionalidade correta do método *getUserFloat*. Ao garantir que o método seja capaz de capturar e processar valores *float* de forma precisa, contribuímos para a melhoria da experiência do usuário e para a integridade dos dados manipulados pelo sistema. Este teste de unidade é essencial para garantir a qualidade do sistema e sua capacidade de lidar com entradas do usuário de maneira eficiente.

### **D. Título: Teste de Unidade para Entrada de Escolha do Usuário**

#### **Introdução:**

O teste de unidade desempenha um papel essencial na validação da funcionalidade correta de métodos que lidam com a entrada de escolha do usuário. Neste contexto, o teste tem como objetivo verificar a precisão da captura de uma escolha numérica feita pelo usuário por meio do método *getUserChoice* em um determinado sistema.

#### **Descrição do Teste:**

O teste concentra-se na simulação de entrada de uma escolha numérica específica feita pelo usuário. Ele verifica se a escolha numérica capturada corresponde à escolha esperada, considerando os limites inferiores e superiores predefinidos.

#### **Passos do Teste:**

- 1) Simular a entrada de uma escolha numérica específica.

- 2) Capturar a escolha numérica feita pelo usuário por meio do método *getUserChoice*.
- 3) Verificar se a escolha capturada corresponde à escolha esperada dentro dos limites definidos.

#### **Resultados Esperados:**

Espera-se que o teste valide a precisão da captura de escolhas numéricas feitas pelo usuário, garantindo que o sistema possa processar corretamente as escolhas inseridas.

#### **Critérios de Sucesso e Falha:**

*Sucesso:* A escolha numérica capturada corresponde exatamente à escolha esperada dentro dos limites definidos.

*Falha:* Qualquer divergência entre a escolha capturada e a escolha esperada, ou se a escolha estiver fora dos limites predefinidos, indica um possível problema na captura de escolhas do usuário.

#### **Observações e Considerações:**

É crucial garantir que a simulação de entrada de dados seja realizada de forma precisa para fornecer a escolha numérica desejada ao método de teste. Além disso, os limites predefinidos devem ser cuidadosamente verificados para garantir que o método *getUserChoice* restrinja corretamente as escolhas de entrada conforme necessário.

#### **Conclusão:**

O teste de unidade para a captura de escolhas do usuário desempenha um papel vital na validação da funcionalidade correta do método *getUserChoice*. Ao garantir que o método seja capaz de capturar e processar escolhas do usuário de forma precisa, contribuímos para a melhoria da interatividade do sistema e para a integridade das operações realizadas pelo sistema. Este teste de unidade é essencial para assegurar a qualidade do sistema e sua capacidade de lidar com escolhas do usuário de maneira eficaz.

*E. Título: Teste de Unidade para Entrada de Data pelo Usuário*

#### **Introdução:**

O teste de unidade é fundamental para validar a funcionalidade correta de métodos que lidam com a entrada de data do usuário. Neste contexto, o teste visa verificar a precisão da captura de uma data fornecida pelo usuário por meio do método *getDate* em um sistema específico.

#### **Descrição do Teste:**

O teste concentra-se na simulação da entrada de uma data específica fornecida pelo usuário. Ele verifica se a data capturada corresponde à data esperada, levando em consideração o formato de data predefinido.

#### **Passos do Teste:**

- 1) Simular a entrada de uma data específica.
- 2) Capturar a data inserida pelo usuário por meio do método *getDate*.
- 3) Verificar se a data capturada corresponde à data esperada de acordo com o formato de data predefinido.

#### **Resultados Esperados:**

Espera-se que o teste valide a precisão da captura de datas fornecidas pelo usuário, garantindo que o sistema possa processar corretamente as datas inseridas.

#### **Critérios de Sucesso e Falha:**

*Sucesso:* A data capturada corresponde exatamente à data esperada, de acordo com o formato de data predefinido.

*Falha:* Qualquer divergência entre a data capturada e a data esperada, ou se a data não estiver no formato correto, indica um possível problema na captura de datas pelo usuário.

#### **Observações e Considerações:**

É crucial garantir que a simulação de entrada de dados seja feita com precisão para fornecer a data desejada ao método de teste. Além disso, o formato de data predefinido deve ser verificado para garantir que o método *getDate* processe corretamente as datas de acordo com o formato especificado.

#### **Conclusão:**

O teste de unidade para a captura de datas pelo usuário desempenha um papel crucial na validação da funcionalidade correta do método *getDate*. Ao garantir que o método seja capaz de capturar e processar datas de forma precisa, contribuímos para a melhoria da funcionalidade do sistema e para a integridade das operações relacionadas a datas realizadas pelo sistema. Este teste de unidade é essencial para garantir a qualidade do sistema e sua capacidade de lidar com entradas de datas do usuário de maneira eficiente.

#### **REFERÊNCIAS**

- 1) SILVA FILHO, R. L. L. E. et al.. *A evasão no ensino superior brasileiro. Cadernos de Pesquisa*, v. 37, n. 132, p. 641–659, set. 2007.
- 2) Nicol, D. J., Macfarlane-Dick, D. (2006). *Formative assessment and self-regulated learning: A model and seven principles of good feedback practice*.
- 3) JUNIT. *JUnit 5 User Guide*. Disponível em: <https://junit.org/junit5/docs/current/user-guide/>. Acesso em: 19 out. 2023.