INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE MINAS GERAIS (IFMG) - CAMPUS BAMBUÍ
BACHARELADO EM ENGENHARIA DA COMPUTAÇÃO

Arthur Albino Resende
Daiana Aparecida Garcia Oliveira

TRABALHO PRÁTICO : **Sistema de memorização espaçada**

Bambuí
2024

ARTHUR ALBINO RESENDE
DAIANA APARECIDA GARCIA OLIVEIRA

TRABALHO PRÁTICO : **Sistema de memorização espaçada**

Trabalho Prático apresentado ao Curso Bacharelado em Engenharia da Computação do Instituto Federal de Minas Gerais - *Campus* Bambuí para aprovação na disciplina.
Professor : Felipe Lopes de Melo Faria

Bambuí
2024

SUMÁRIO

INTRODUÇÃO

Este relatório detalha o desenvolvimento de um sistema para a empresa fictícia Learn, especializada no ensino de línguas estrangeiras para pessoas de todas as idades e gêneros. A empresa busca melhorar a retenção do conteúdo estudado por seus alunos por meio de um sistema de memorização espaçada.

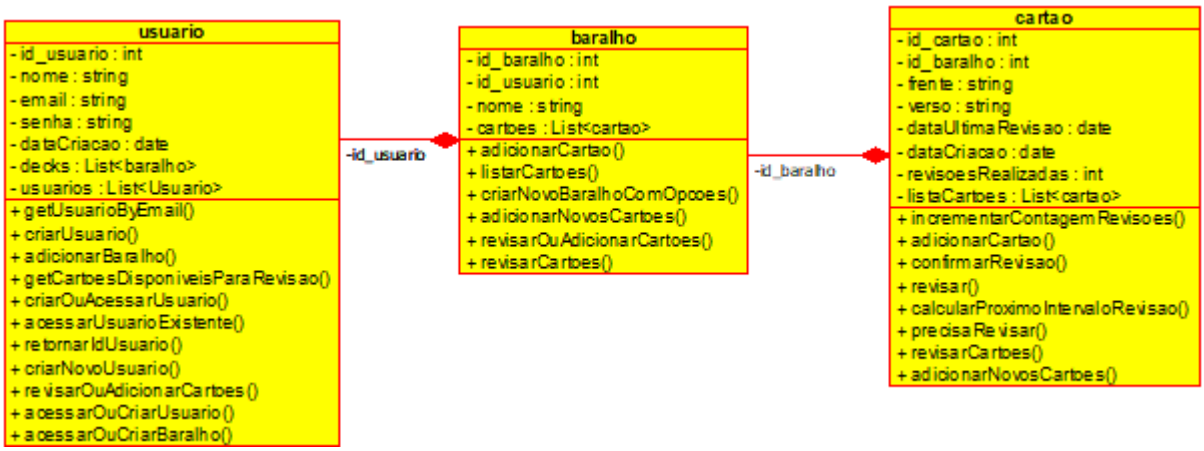
O sistema de repetição espaçada é um método de memorização fundamentado na curva do esquecimento de *Ebbinghaus*. Desde sua criação, tem sido reconhecido como um dos métodos mais eficazes para o aprendizado de vocabulário. Esse método baseia-se no princípio, proposto por *Ebbinghaus*, de que uma informação deve ser revisada regularmente para ser consolidada permanentemente em nossa memória.

O sistema permite o registro de múltiplos usuários, cada um capaz de criar e gerenciar seus próprios baralhos e cartões. Além disso, facilita a revisão dos cartões em intervalos programados, como imediatamente após sua criação e posteriormente após 1, 7, 30 dias, e assim por diante. Para a persistência dos dados, optamos pelo banco de dados PostgreSQL.

DESENVOLVIMENTO

Na fase de planejamento, foram definidos os principais objetivos do sistema e identificados os requisitos fundamentais, como cadastro de usuários, criação e gerenciamento de baralhos e cartões, e revisão dos cartões em intervalos programados. Com base nesses requisitos, foi desenvolvido o seguinte diagrama de classes:

Figura 1 : Diagrama de classes



O diagrama de classes foi composto por três entidades principais: Usuário, Baralho e Cartão. Este diagrama serviu como uma representação visual da estrutura fundamental do sistema, delineando as relações e interações entre essas entidades.

A classe usuario representa os indivíduos registrados no sistema, cada um dos quais possui a capacidade de criar e gerenciar seus próprios baralhos de cartões. Por sua vez, a classe baralho está intimamente relacionada ao usuario por meio de um relacionamento do tipo composição, refletindo o fato de que cada usuário pode possuir vários baralhos contendo diferentes conjuntos de cartões. Dentro de cada baralho, os cartões são armazenados como instâncias da classe cartão. Esses cartões podem conter informações como palavras, frases, perguntas ou qualquer outro tipo de conteúdo relevante para o propósito do sistema. A relação entre baralho e cartão também é de composição, indicando que cada baralho pode conter múltiplos cartões, os quais são exclusivos desse baralho específico. Além disso, o diagrama de classes contemplou outros atributos e métodos relevantes para cada entidade, como métodos

para adicionar, criar e revisar cartões dentro de um baralho, assim como também métodos para criar usuários e baralhos.

A classe usuario possui atributos como id_usuario, nome, email, senha, dataCriacao, e decks. O id_usuario é utilizado como identificador único de cada usuário no sistema. Para autenticação, o e-mail e a senha são utilizados. Cada usuário pode cadastrar seus próprios baralhos (decks).

Entre os principais métodos da classe usuario, destacam-se os seguintes:

criarNovoUsuario(): Este método permite a criação de um novo usuário. São necessários o fornecimento do nome, e-mail e senha desejados. O id de usuário é gerado pelo banco de dados.

acessarUsuarioExistente(): Permite que um usuário existente acesse sua conta. Para isso, o usuário deve inserir seu e-mail e senha. O método realiza a busca no banco de dados e valida o acesso.

revisarOuAdicionarCartoes(): Este método busca os baralhos do usuário, permitindo revisar os cartões existentes ou adicionar novos. A busca é realizada no banco de dados por meio de outro método.

A classe baralho possui atributos como id_baralho, nome e cartoes, onde o id_baralho é gerado automaticamente pelo banco de dados, e o id_usuario é obtido da classe usuario.

Entre os principais métodos da classe baralho, destacam-se os seguintes:

criarNovoBaralhoComOpcoes(): Este método possibilita a criação de novos baralhos. Requer autenticação do usuário e solicita o nome desejado para o novo baralho. Após a inserção no banco de dados, obtém o id_baralho gerado.

revisarCartoes(): Recebe como parâmetro o baralho escolhido e a lista de cartões, permitindo ao usuário revisar seus cartões.

A classe cartao possui diversos atributos essenciais, como id_baralho, id_cartao, frente, verso, dataUltimaRevisao, dataCriacao, revisoesRealizadas e listaCartoes. O

id_baralho é derivado da classe baralho, enquanto o id_cartao é gerado automaticamente pelo banco de dados. A frente e o verso armazenam respectivamente o conteúdo frontal e traseiro do cartão. Os atributos dataUltimaRevisao e revisoesRealizadas são cruciais para o cálculo da próxima revisão do cartão.

Dentre os principais métodos da classe cartao, destacam-se os seguintes:

incrementarContagemRevisoes(): Este método incrementa o contador de revisões realizadas pelo cartão.

adicionarCartao(): Responsável por solicitar ao usuário as informações de frente e verso do cartão. Obtém do banco de dados o id_baralho por meio de outro método e, com auxílio de um método adicional, salva o cartão no banco de dados.

calcularProximoIntervaloRevisao(): Realiza o cálculo do próximo intervalo de revisão. Baseia-se na contagem atual de revisões do cartão: se for a primeira revisão, a próxima será em 1 dia; se for a segunda, em 7 dias; e para três ou mais revisões, será após 30 dias.

precisaRevisar(): Verifica se o cartão precisa ser revisado. Se nunca foi revisado, será selecionado para revisão. Além disso, se a soma do intervalo de revisão calculado com a data da última revisão for igual ou anterior ao dia atual, o cartão será marcado para revisão.

Além das classes principais, foram desenvolvidas classes auxiliares essenciais para garantir o correto funcionamento do sistema, especialmente no que diz respeito ao gerenciamento do banco de dados. Essas classes desempenham um papel fundamental ao fornecer uma interface eficiente para acessar e manipular os dados armazenados no banco de dados. Elas facilitam operações como inserção, atualização, exclusão e consulta de informações, garantindo a integridade e a consistência dos dados em todo o sistema. Com o auxílio dessas classes auxiliares, tornou-se possível uma interação eficaz entre a lógica do programa e o banco de dados subjacente, garantindo um desempenho otimizado e uma experiência confiável para os usuários do sistema.

A classe user001 oferece os seguintes métodos principais para o gerenciamento do banco de dados:

salvarUsuario(): Recebe um objeto usuário contendo os atributos nome, email, senha e data de criação, e os armazena no banco de dados.

getUsuarioByEmailSenha(): Recebe um email e senha como entrada e, se existirem no banco de dados, retorna o id do usuário, data de criação e nome associados.

A classe card001 apresenta os seguintes métodos principais:

buscarCartoes(): Realiza uma busca no banco de dados e retorna uma lista de objetos cartão contendo os atributos id_cartao, frente, verso, data de última revisão, data de criação e revisões realizadas.

inserirCartaoNoBancoDeDados(): Recebe o id do baralho e um objeto cartão preenchido com informações como frente, verso, data de criação e id do baralho, e os armazena no banco de dados.

atualizarDataUltimaRevisaoNoBancoDeDados(): Recebe um objeto cartão e atualiza a data de última revisão associada a ele.

buscarCartoesPorBaralho(): Recebe o id do baralho como entrada e retorna os cartões relacionados a ele.

Por fim, a classe desks001 conta com os seguintes métodos principais de gerenciamento no banco de dados:

inserirBaralhoNoBancoDeDados(): Recebe o id do usuário e o nome do baralho como entrada, insere as informações no banco de dados e obtém o id do baralho gerado.

acessarBaralhoExistente(): Recebe um objeto usuário como entrada e retorna os baralhos associados ao seu id.

CONCLUSÃO

A abordagem baseada em Orientação a Objetos desempenhou um papel crucial em cada etapa do projeto. Desde a fase inicial de planejamento, onde foram identificados os requisitos principais e definidos os objetivos do sistema, até a implementação das classes principais como Usuário, Baralho e Cartão, a orientação a objetos proporcionou uma estrutura sólida e flexível para o desenvolvimento do sistema.

A utilização das classes auxiliares para o gerenciamento do banco de dados, como user001 e card001, demonstrou como a orientação a objetos facilita a organização e a manutenção do código, além de garantir a integridade e a consistência dos dados em todo o sistema.

Os métodos principais implementados em cada classe, como salvarUsuario(), buscarCartoes() e inserirBaralhoNoBancoDeDados(), exemplificam a eficácia da abordagem orientada a objetos na implementação de funcionalidades específicas do sistema, possibilitando uma interação eficiente com o banco de dados e proporcionando uma experiência confiável para os usuários.

Portanto, concluímos que a orientação a objetos foi fundamental para o sucesso do projeto, permitindo a criação de um sistema de memorização espaçada eficiente, escalável e adaptável .

REFERÊNCIAS

Mosalingua. **O Sistema de Repetição Espaçada (SRS):** memorizar para jamais esquecer. Disponível em: https://www.mosalingua.com/pt/o-sistema-de-repeticao-espacada-memorizar-para-jamais-esquecer/. Acesso em: 16 maio 2024.

Blogdoead. **Conheça a Repetição Espaçada, método que vai turbinar seus estudos**. Disponível em: https://www.blogdoead.com.br/tag/metodologias-de-estudo/repeticao-espacada. Acesso em: 16 maio 2024.

Basmo. **Conheça a Repetição Espaçada, método que vai turbinar seus estudos**. Disponível em: https://www.blogdoead.com.br/tag/metodologias-de-estudo/repeticao-espacada. Acesso em: 16 maio 2024.

Essential. **Sistema de Repetição Espaçada para Ensinar e Aprender um Idioma**. Disponível em: https://essential.com.br/sistema-de-repeticao-espacada/. Acesso em: 16 maio 2024.

APÊNDICE

Link do projeto no GitHub: https://github.com/daianagarcia2805/Programa-o-Orientada-a-Objetos