

COMO UM APLICATIVO DE GESTÃO ACADÊMICA PODE REDUZIR A EVASÃO ESCOLAR NO ENSINO SUPERIOR

HOW AN ACADEMIC MANAGEMENT APPLICATION CAN REDUCE DROPOUTS IN HIGHER EDUCATION

Alexandre José Peixoto dos Santos – Alexandre3091@outlook.com
 Faculdade de Tecnologia de Taquaritinga – Taquaritinga – São Paulo – Brasil

Giuliano Sombatti Pinto – giuliano.pinto@fatec.sp.gov.br
 Faculdade de Tecnologia de Taquaritinga – Taquaritinga – São Paulo – Brasil

DOI: 10.31510/infa.v21i2.2055
 Data de submissão: 24/09/2024
 Data do aceite: 23/11/2024
 Data da publicação: 20/12/2024

RESUMO

Este estudo aborda o desafio de conciliar a vida acadêmica e pessoal, o que frequentemente resulta em altas taxas de evasão escolar no ensino superior. Propõe-se o desenvolvimento de um aplicativo móvel de gestão acadêmica com o objetivo de melhorar a organização dos alunos e reduzir a evasão escolar. O principal objetivo é descrever o processo de desenvolvimento do aplicativo, que oferece funcionalidades como acompanhamento de prazos, controle de frequência e alertas personalizados para aumentar a permanência dos estudantes. Foi realizada uma revisão bibliográfica para identificar os principais requisitos, seguida da criação de protótipos de média fidelidade e da modelagem do sistema com diagramas UML. A codificação foi feita em Java no ambiente Android Studio. Os resultados indicam que funcionalidades como notificações personalizadas, controle de frequência e agendamento de atividades podem apoiar a organização dos alunos e diminuir o risco de evasão. Apesar de o desenvolvimento do aplicativo ainda estar em andamento, ele apresenta grande potencial para melhorar a gestão acadêmica. Os próximos passos incluem testes em ambientes reais para validar suas funcionalidades e interface.

Palavras-chave: Evasão escolar. Gestão Acadêmica. Aplicativo. Ensino Superior. Tecnologia.

ABSTRACT

This study addresses the challenge of reconciling academic and personal life, which often results in high dropout rates in higher education. It is proposed to develop an academic management mobile application with the aim of improving student organization and reducing school dropout rates. The main objective is to describe the application development process, which offers features such as deadline monitoring, attendance control and personalized alerts to increase student retention. A literature review was carried out to identify the main requirements, followed by the creation of medium fidelity prototypes and modeling of the system with UML diagrams. The work was done in Java in the Android Studio environment.

The results indicate that features such as personalized notifications, attendance control and activity scheduling can support the organization of classes and reduce the risk of dropout. Although the development of the application is still ongoing, it has great potential to improve academic management. The next steps include testing in real environments to validate its functionalities and interface.



Keywords: Dropout. Academic Management. Application. Higher Education. Technology.

1 INTRODUÇÃO

A conciliação entre vida pessoal e acadêmica é desafiadora, principalmente para estudantes do ensino superior, especialmente àqueles que buscam um equilíbrio entre o aprimoramento educacional e a inserção no mercado de trabalho. Esse desequilíbrio muitas vezes leva à sobrecarga e à dificuldade de organização, resultando em altos índices de evasão escolar. Como destacado por Azevedo (2006), a evasão é uma preocupação crescente, que traz prejuízos não apenas na trajetória educacional dos alunos, mas também na sociedade como a desestruturação de famílias, falta de empregabilidade e outros prejuízos sociais e pessoais.

Visando enfrentar essa problemática, o presente trabalho propõe o desenvolvimento de um aplicativo móvel para gestão acadêmica, com o objetivo de melhorar a organização e o acompanhamento das atividades dos alunos, ajudando a reduzir a evasão escolar. O aplicativo oferece funcionalidades como acompanhamento de prazos, controle de frequência e envio de alertas personalizados, visando atender às necessidades específicas dos estudantes de ensino superior. Dessa forma, espera-se proporcionar uma ferramenta que auxilie os alunos a equilibrem suas responsabilidades acadêmicas e pessoais, reduzindo o risco de abandono escolar.

O objetivo deste artigo é descrever as etapas de desenvolvimento do aplicativo e discutir como ele pode contribuir para a melhoria da gestão acadêmica, destacando as principais decisões e desafios enfrentados ao longo do processo. Adicionalmente, este estudo levanta a hipótese de que a implementação de ferramentas de gestão mais acessíveis e personalizáveis pode resultar em um aumento na permanência dos estudantes no ensino superior.

Esse artigo está organizado da seguinte forma: na seção 2, será apresentada a fundamentação teórica, discutindo a evasão escolar e o uso da tecnologia na educação. Na seção 3, serão descritos os procedimentos metodológicos adotados para o desenvolvimento do

aplicativo. A seção 4 aborda os resultados obtidos durante o processo de desenvolvimento, seguido pela seção 5, que traz as considerações finais, limitações do estudo e sugestões para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Evasão Escolar no Ensino Superior

A evasão escolar no ensino superior é um desafio significativo para estudantes e a sociedade. Segundo Azevedo (2006), ela impacta o desenvolvimento individual e coletivo, dificultando a inserção dos jovens no mercado de trabalho e sua estabilização financeira. Esses efeitos destacam a importância de enfrentar o problema de forma eficaz.

No Brasil, a evasão escolar é persistente. De acordo com o Instituto Brasileiro de Geografia e Estatística (IBGE) (2020), cerca de 10,1 milhões de jovens não completaram as etapas básicas da educação. Apesar de transformações no sistema educacional, o país ainda lida com problemas como o analfabetismo e a evasão escolar, conforme apontado por Azevedo (2006). O Censo da Educação Superior realizado pelo Instituto Nacional de Estudos e Pesquisas Educacionais (INEP) (2019), revelou que, em 2010, 56,8% dos estudantes que ingressaram no ensino superior desistiram do curso, e apenas 37,9% conseguiram se formar, o que evidencia a gravidade da situação.

A evasão escolar é um fenômeno multifatorial. Ferreira e Oliveira (2020) afirmam que ela pode prejudicar não apenas o aluno, mas também a sociedade, ao excluir o indivíduo de uma realidade letrada e economicamente ativa. O IBGE (2020) identificou as principais razões para a evasão entre jovens de 14 a 29 anos: a necessidade de trabalhar (39,1%) e a falta de interesse (29,2%), refletindo a difícil realidade socioeconômica enfrentada por muitos brasileiros.

Além disso, a ausência de ferramentas adequadas para auxiliar na gestão acadêmica dos estudantes contribui para o aumento da evasão, uma vez que questões como a organização e o cumprimento de prazos tornam-se mais difíceis de gerenciar. A fim de enfrentar essas dificuldades, torna-se necessária a criação de soluções tecnológicas que possam apoiar os alunos de forma eficaz. Com isso em mente, propõe-se o desenvolvimento de um aplicativo com funcionalidades que supram as necessidades dos estudantes.

2.2 Tecnologia na Educação

A tecnologia tem desempenhado um papel cada vez mais importante na transformação da educação. A incorporação de ferramentas digitais e recursos tecnológicos nas salas de aula têm revolucionado a forma como os alunos aprendem e os professores ensinam (Escola Educação, 2023). Entre essas ferramentas, existem as voltadas à gestão acadêmica, normalmente controladas pelos professores. Um exemplo disso são os sistemas utilizados em instituições de ensino superior, nos quais os orientadores têm o controle exclusivo de funções como alteração de notas, registro de frequência e definição do status de aprovação ou reaprovação dos alunos.

Embora esses sistemas cumpram seus objetivos, eles podem se tornar entediantes e desmotivadoras para os estudantes devido à falta de autonomia. Ao oferecer apenas a possibilidade de acompanhar a própria situação, não permitem que o aluno realize ações ativas e, desse modo, esses sistemas limitam o envolvimento. Essa sensação de falta de controle é explicada pela Teoria da Autodeterminação, proposta por Deci e Ryan (2000), que argumentam que ao não proporcionar um ambiente que permita que as pessoas sejam capazes, independentes e conectadas aos outros, essas instituições acabam por gerar um sentimento de desconexão e insatisfação. Isso evidencia a necessidade de um maior nível de interação e controle por parte dos alunos, tornando a gestão acadêmica mais envolvente e eficiente.

Na busca de uma plataforma acessível para o desenvolvimento da aplicação, foram investigadas as tecnologias mais prevalentes, sendo a principal delas os dispositivos móveis, como smartphones, que desempenham um papel central. De acordo com um estudo da FGV (2022), existem mais de 249 milhões de celulares em uso no Brasil. O sistema operacional Android, está presente em 82% desses dispositivos segundo pesquisa da StatCounter (2024). Dessa forma, o Android destaca-se como uma plataforma ideal para o desenvolvimento de aplicativos educacionais. Sua ampla adoção e acessibilidade permitem que mais estudantes tenham acesso a essas soluções, favorecendo o engajamento e a autonomia no processo de aprendizado.

2.3 Seleção de Linguagens de Programação e Ambientes de Desenvolvimento

Deve-se trabalhar para determinar se ele (software) pode ser concluído segundo as dimensões da tecnologia, do orçamento, do tempo e dos recursos disponíveis (Pressman, 2016). Para isso, ao determinar o escopo de um software, é importante considerar e definir diferentes quesitos, entre eles as tecnologias a serem utilizadas.

No caso do desenvolvimento de um aplicativo móvel Android, foram consideradas duas linguagens de programação: Java e Kotlin. Após uma análise cuidadosa, optou-se por utilizar Java, com base em critérios como estabilidade, suporte e longevidade.

Desenvolvida em 1995, Java é uma linguagem de programação orientada a objetos, consolidada no mercado há mais de 20 anos (Microsoft, s.d.). Sua longa trajetória proporciona uma vasta quantidade de bibliotecas, uma grande comunidade de suporte e ampla utilização em diversos contextos, incluindo o desenvolvimento mobile. Além disso, Java é a linguagem nativa para dispositivos Android e é comumente usada na *Integrated Development Environment* (IDE) Android Studio, um ambiente robusto para o desenvolvimento de aplicativos (Android Studio, 2024).

Uma das razões para escolha do Java foi sua estabilidade e confiabilidade, fatores importantes para garantir que o sistema tenha suporte ao longo prazo e seja compatível com futuras adições ou modificações, algo essencial, tendo em mente que o aplicativo não foi finalizado. Embora Java tenha uma curva de aprendizagem mais íngreme, sua popularidade e amplo suporte compensam esse aspecto. Como evidenciado por Barbosa e Bessa (2020), que conduziram um questionário com desenvolvedores, Java foi considerado um pouco mais desafiador que Kotlin, com 30,6% dos votos, em comparação aos 55,6% que consideraram Kotlin mais fácil. Mesmo assim, a escolha pelo Java se justifica pela robustez e maturidade da linguagem, essenciais para garantir um desempenho eficiente e confiável no desenvolvimento do aplicativo.

3 PROCEDIMENTOS METODOLÓGICOS

A metodologia utilizada no presente trabalho foi a pesquisa bibliográfica. Os materiais utilizados para o seu desenvolvimento foram livros, artigos e sites. Com a pesquisa e definição dos principais materiais relevantes relacionados ao tema, foram abordadas as ferramentas utilizadas para protótipo, modelar e desenvolver o software.

A fase inicial envolveu o levantamento de requisitos, através da pesquisa bibliográfica, identificando as principais dificuldades encontradas pelos alunos e utilizando de teorias já estabelecidas para uma possível solução.

Em seguida, foi realizada a criação de protótipos de média fidelidade utilizando o software Figma, escolhido por sua robustez e variedade de recursos. Como destacado por Pressman (2016), os protótipos servem como uma forma de definição de requisitos. Essa definição de requisitos valida os requisitos anteriores e facilita e agiliza o processo de

desenvolvimento. Esses protótipos foram desenvolvidos durante vários momentos, sendo aperfeiçoados e refeitos com o passar do tempo, buscando constantemente aprimorar a experiência do usuário, ao mesmo tempo em que se focavam em aumentar a agilidade dos processos e garantir a simplicidade no uso das funcionalidades.

Após a prototipagem, a modelagem do sistema foi realizada utilizando boas práticas de design, como a aplicação de diagramas UML (*Unified Modeling Language*). A UML como descrito por Booch, Rumbaugh e Jacobson (2012), é uma linguagem padrão para a criação e desenvolvimento de estruturas de software. Para isso, foram criados diagramas de classe para organizar e visualizar as estruturas do sistema. Esses diagramas facilitaram o alinhamento entre as funcionalidades desenvolvidas e as necessidades específicas dos estudantes. Para criar os diagramas, foi utilizado o software draw.io, escolhido por sua facilidade de uso e vasta gama de recursos.

A etapa de codificação foi baseada nos protótipos e na modelagem previamente desenvolvida. O Android Studio foi utilizado como ambiente de desenvolvimento, e a escolha da linguagem de programação foi o Java, pela sua robustez e compatibilidade com a plataforma Android. A codificação seguiu os padrões estabelecidos nas fases anteriores, e cerca de 70% das funcionalidades do sistema foram implementadas, incluindo a parte visual (*front-end*) e a parte lógica (*back-end*).

4 RESULTADOS E DISCUSSÃO

Nesta seção, serão apresentados os resultados obtidos nas diferentes fases do desenvolvimento do aplicativo, incluindo o levantamento de requisitos, prototipagem, a modelagem com diagramas UML e a implementação das principais funcionalidades. Esses resultados refletem as escolhas metodológicas descritas anteriormente e mostram como as decisões tomadas contribuíram para a construção de uma solução eficaz para a gestão acadêmica.

4.1 Levantamento de Requisitos

Para ser eficaz, um aplicativo educacional precisa ser fácil de usar, considerando que será utilizado por um público diversificado. Conforme Norman (2002), "um método importante de tornar os sistemas mais fáceis de aprender e usar é torná-los exploráveis, incentivando o usuário a aprender por meio da exploração ativa". Isso requer atenção a

aspectos como organização visual, personalização e criação de hábitos que promovam engajamento contínuo.

A organização das informações será feita por meio de cartões, que agrupam matérias e faltas, facilitando o processamento rápido e eficiente dos dados. A personalização dos cartões, com cores e ícones ajustáveis, é essencial para aumentar o senso de controle e engajamento dos alunos, como apontado pela pesquisa da McKinsey & Company (2021), que indica que 71% dos usuários esperam interações personalizadas. Isso se alinha à Teoria da Autodeterminação, que destaca a importância da autonomia na motivação pessoal.

A gestão eficiente do tempo também é uma necessidade fundamental. Ferreira e Batista (2011) enfatizam que o tempo é um recurso irreversível, sendo crucial o uso de lembretes e notificações para otimizar a organização pessoal. O aplicativo permitirá a configuração de alertas personalizados, que notificarão o usuário sobre prazos ou faltas acumuladas, contribuindo para uma gestão do tempo mais eficaz, conforme Eyal (2020).

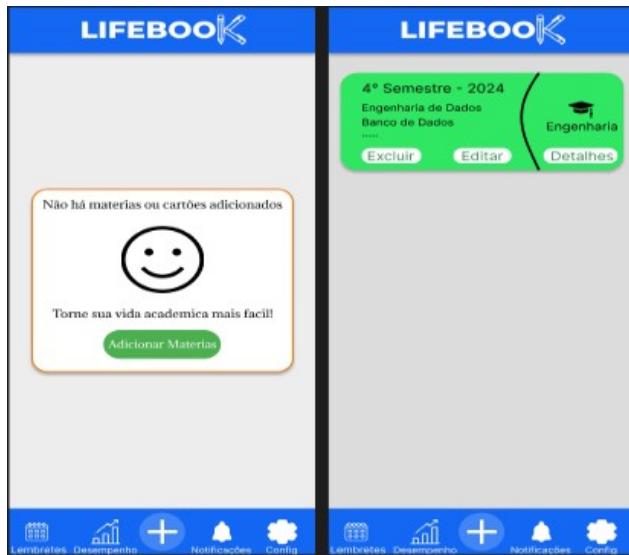
Por fim, a implementação de gráficos facilitará a visualização do desempenho acadêmico, tornando a gestão de faltas mais intuitiva e menos estressante, como observado por Corrêa (2019).

4.2 Protótipos

De acordo com Pressman (2016), os desenvolvedores ou modeladores não devem apegar-se aos protótipos, visto que são realizados de forma iterativa e não tendo como objetivo ser a versão final do software (na maior parte dos casos), mas sim como uma garantia de qualidade. Com isso em mente, serão abordados os protótipos finais desenvolvidos.

Na figura 1 a seguir, pode-se observar duas versões da tela inicial do aplicativo. Quando não há um cartão adicionado, o aplicativo exibe uma mensagem para incentivar a criação, mantendo um *layout* simples e organizado para facilitar o gerenciamento acadêmico. Já na segunda versão, com um cartão adicionado, o aplicativo exibe o cartão com três opções de gerenciamento: edição, exclusão e visualização de detalhes. Além disso, o cartão exibe as informações relevantes, como o nome do cartão e as principais matérias vinculadas. Em ambas as telas, há uma faixa de opções na parte inferior, disponível em todas as telas, que permite fácil acesso a outras funcionalidades do sistema.

Figura SEQ Figura * ARABIC 1 - (Esquerda) Tela inicial sem cartão adicionado; (Direita) Tela com cartão adicionada.



Fonte: Elaborado Pelo Autor (2024)

4.3 Diagramas de Classe

O diagrama de classe foi elaborado para representar as principais funcionalidades do sistema e suas relações. Segundo Sommerville (2013), esses diagramas modelam sistemas orientados a objetos, exibindo as classes e suas associações. Cada classe contém atributos (nome e tipo), operações (métodos) e visibilidade indicada pelos símbolos "+" (público) e "-" (privado). As classes se conectam através de setas que indicam seus relacionamentos.

Os cartões, que são a base do sistema, armazenam informações como id, nome, ícone escolhido, cor do texto, cor do botão e cor do cartão. Um cartão pode ter várias matérias associadas. As funcionalidades de adição, edição e exclusão são essenciais, exigindo dados como nome do cartão, ícone e cores.

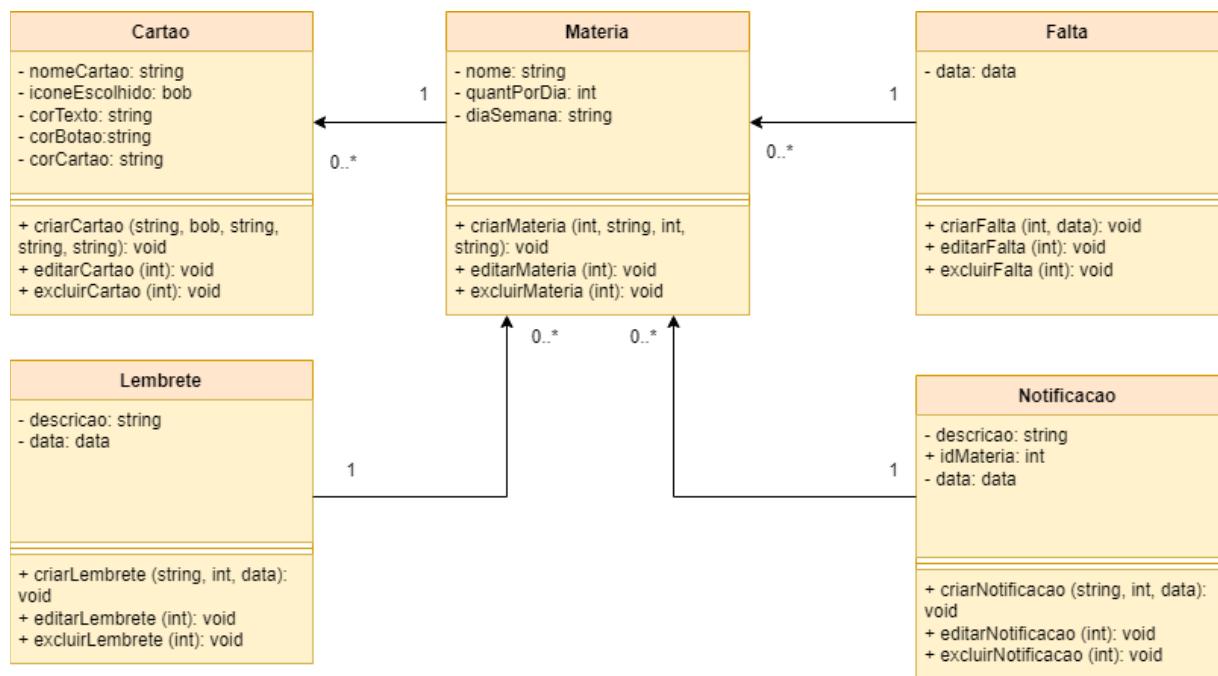
As matérias, vinculadas aos cartões, contêm id, id do cartão, nome, quantidade de aulas por dia e dia da semana. Cada matéria pertence a um único cartão, e na adição de uma matéria, são requeridos o id do cartão, nome, quantidade de aulas por dia e dia da semana.

Além disso, podem ser adicionadas faltas às matérias, que contêm id, id da matéria e data. Cada falta pode estar associada a uma matéria e pode ser adicionada, editada ou excluída conforme necessário.

Lembretes e notificações dependem das matérias criadas, contendo id, descrição, id da matéria e data. Na adição de lembretes, são fornecidos a descrição, id da matéria e a data, enquanto edição e exclusão exigem apenas o id do lembrete.

A Figura 2 ilustra as conexões entre essas funcionalidades, detalhando os parâmetros e funções principais.

Figura SEQ Figura * ARABIC 2 - Diagrama de Classe



Fonte: Elaborado Pelo Autor (2024)

4.4 Codificação

Na fase de codificação, foi implementada uma série de funcionalidades para a gestão visual e lógica dos cartões no aplicativo. A seguir estão alguns exemplos de trechos de código relevantes para o desenvolvimento.

A figura 3 a seguir, demonstra uma função responsável por criar um adaptador personalizado para exibir cores em uma lista. A classe color adapter utiliza a classe GradientDrawable para desenhar um quadrado com a cor fornecida. Neste caso, o método getView() infla a View apenas quando necessário, garantindo uma performance otimizada. Este código define visualmente as cores dos cartões, permitindo ao usuário personalizar a aparência do aplicativo e proporcionar uma experiência mais intuitiva.

Figura SEQ Figura * ARABIC 3 - Função para Alterar Cor do Cartão

```

@Override
public View getView(int position, View convertView, ViewGroup parent) {
    View view = convertView;
    if (view == null) {
        LayoutInflator inflater = (LayoutInflator) context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        view = inflater.inflate(R.layout.color_list_item, null);
    }

    ImageView colorSquare = view.findViewById(R.id.colorSquare);
    GradientDrawable drawable = new GradientDrawable();
    drawable.setColor(Color.parseColor(hexColors[position]));
    drawable.setShape(GradientDrawable.RECTANGLE);
    drawable.setSize(70, 70); // Ajuste o tamanho conforme necessário
    colorSquare.setImageDrawable(drawable);

    return view;
}

```

Fonte: Elaborado Pelo Autor (2024)

A figura 4 a seguir, representa uma função da classe Agrupamento, mais especificamente, a função responsável por buscar um agrupamento em específico através de seu código único. Com o código único do agrupamento, o sistema realiza uma consulta SQL para recuperar as informações do agrupamento, como nome, categoria e as cores de fundo, texto e botões.

Figura SEQ Figura * ARABIC 4 - Função Responsável por Buscar um Agrupamento

```

Fonte: https://github.com/ElaboradoPeloAutor/Projeto-Aplicativo-Movel-para-Estudantes-Android/blob/main/app/src/main/java/com/example/projetoaplicativomovel/dao/AgrupamentoDAO.java#L14-L75

private Agrupamento obterAgrupamentoPorId(int agrupamentoId) {
    List<Agrupamento> agrupamentos = new ArrayList<>();

    try {
        bancoDados = dbHelper.getReadableDatabase();
        Cursor cursor = bancoDados.rawQuery("SELECT * FROM agrupamento WHERE id_agrupamento = ?", new String[]{String.valueOf(agrupamentoId)});

        if (cursor != null && cursor.moveToFirst()) {
            int nomeAgrupamentoIndex = cursor.getColumnIndex("nome_agrupamento");
            int categoriaIndex = cursor.getColumnIndex("categoria");
            int corFundoHexIndex = cursor.getColumnIndex("corFundoHex");
            int corTextoHexIndex = cursor.getColumnIndex("corTextoHex");
            int corBotõesHexIndex = cursor.getColumnIndex("corBotõesHex");

            if (nomeAgrupamentoIndex >= 0 && categoriaIndex >= 0 && corFundoHexIndex >= 0 && corTextoHexIndex >= 0 && corBotõesHexIndex >= 0) {
                String nomeAgrupamento = cursor.getString(nomeAgrupamentoIndex);
                String categoria = cursor.getString(categoriaIndex);
                String corFundoHex = cursor.getString(corFundoHexIndex);
                String corBotõesHex = cursor.getString(corBotõesHexIndex);
                String corTextoHex = cursor.getString(corTextoHexIndex);

                return new Agrupamento(nomeAgrupamento, categoria, corFundoHex, agrupamentoId, corTextoHex, corBotõesHex);
            } else {
                Log.e("TAG", "Índice de coluna inválido");
            }
        }
        bancoDados.close();
    }
}

```

CONCLUSÃO

O presente trabalho discorreu sobre as etapas de desenvolvimento de um aplicativo móvel para auxiliar estudantes do ensino superior na gestão de suas atividades acadêmicas, com o intuito de reduzir a evasão escolar. Apesar de o aplicativo ainda não ter sido concluído, as funcionalidades já desenvolvidas demonstram um potencial significativo para melhorar a organização e o acompanhamento das atividades dos alunos.

Com base nas teorias de interação e experiência do usuário, espera-se que o uso de funcionalidades como notificações personalizadas, controle de frequência e agendamento de atividades possa facilitar a conciliação entre vida acadêmica e pessoal, reduzindo o estresse e promovendo uma maior permanência dos alunos em suas trajetórias acadêmicas.

No entanto, é necessário que o aplicativo seja testado em um ambiente real para validar as funcionalidades, interfaces e os benefícios esperados. Pesquisas futuras poderão focar na realização desses testes e ajustes baseados no *feedback*, permitindo avaliar o impacto direto na gestão acadêmica e na redução da evasão escolar.

Dessa forma, o desenvolvimento deste aplicativo contribui tanto para o campo teórico, ao utilizar metodologias de interação e experiência do usuário, quanto para o campo prático, oferecendo uma solução inovadora para os desafios enfrentados pelos estudantes no ensino superior.

REFERÊNCIAS

ANDROID STUDIO. Conheça o Android Studio. Disponível em: <<https://developer.android.com/studio/intro?hl=pt-br>>. Acesso em: 1 abr. 2024.

ARORA, N. et al. The value of getting personalization right—or wrong—is multiplying. Disponível em: <<https://www.mckinsey.com/capabilities/growth-marketing-and-sales/our-insights/the-value-of-getting-personalization-right-or-wrong-is-multiplying>>. Acesso em: 20 set. 2024.

AZEVEDO, F. V. M. Causas e consequências da evasão escolar no ensino de jovens e adultos na escola municipal “Expedito Alves” –Angicos/RN. Dominium, Natal, v. 1, p. 1-38, 2006.

BARBOSA, V. H. F.; BESSA, G. M. A. UM ESTUDO COMPARATIVO SOBRE AS LINGUAGENS JAVA E KOTLIN PARA O DESENVOLVIMENTO DE APLICATIVOS ANDROID. 2020. Trabalho de Conclusão de Curso (Bacharelado em Sistemas de Informação) – Centro Universitário Academia – UniAcademia, 2020. Disponível em: <https://seer.uniacademia.edu.br/index.php/cesi/article/view/2554>. Acesso em: 20 set. 2024.

BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. UML: Guia do usuário. Rio de Janeiro: Elsevier, 2012.

BRASIL. Instituto nacional de estudos e Pesquisas Educacionais Anísio Teixeira (INEP). Censo da educação superior 2018: notas estatísticas. Brasília, 2019. Disponível em: <https://download.inep.gov.br/educacao_superior/censo_superior/documentos/2019/censo_da_educacao_superior_2018-notas_estatisticas.pdf>. Acesso em 2 ago. 2024.

IBGE. Necessidade de trabalhar e desinteresse são principais motivos para abandono escolar. Disponível em: <<https://agenciadenoticias.ibge.gov.br/agencia-noticias/2012->

agencia-de-noticias/noticias/28286-necessidade-de-trabalhar-e-desinteresse-sao-principais-motivos-para-abandono-escolar>. Acesso em: 10 set. 2024.

EDUCAÇÃO, E. O impacto da tecnologia na educação: tendências e perspectivas. Disponível em: <<https://escolaeducacao.com.br/o-impacto-da-tecnologia-na-educacao-tendencias-e-perspectivas/>>. Acesso em: 15 set. 2024.

EYAL, N. Hooked (Engajado): Como construir produtos e serviços formadores de hábitos. 1. ed. São Paulo: AlfaCon, 2020.

FERREIRA, E. C. da S.; OLIVEIRA, N. M. de. EVASÃO ESCOLAR NO ENSINO MÉDIO: causas e consequências. Scientia Generalis, [S. l.], v. 1, n. 2, p. 39–48, 2020. Disponível em: <https://scientiageneralis.com.br/index.php/SG/article/view/v1n2a4>. Acesso em: 20 set. 2024.

FGV. Uso de TI no Brasil: País tem mais de dois dispositivos digitais por habitante, revela pesquisa. Disponível em: <<https://portal.fgv.br/noticias/uso-ti-brasil-pais-tem-mais-dois-dispositivos-digitais-habitante-revela-pesquisa>>. Acesso em: 10 set. 2024b.

LIMA, M. do C. F.; JESUS, S. B. ADMINISTRAÇÃO DO TEMPO: UM ESTUDO SOBRE A GESTÃO EFICAZ DO TEMPO COMO FERRAMENTA PARA O AUMENTO DA PRODUTIVIDADE E WORK LIFE BALANCE. Revista de Gestão e Secretariado, [S. l.], v. 2, n. 2, p. 121–144, 2012. DOI: 10.7769/gesec.v2i2.48. Disponível em: <https://ojs.revistagesec.org.br/secretariado/article/view/48>. Acesso em: 20 set. 2024.

MICROSOFT. O que é o Java?: Um guia para iniciantes em Java | Microsoft Azure. Disponível em: <<https://azure.microsoft.com/pt-br/resources/cloud-computing-dictionary/what-is-java-programming-language>>. Acesso em: 20 set. 2024.

NORMAN, D. A. O Design do dia a dia. Rio de Janeiro: Rocco, 2002.

PRESSMAN, R. S.; MAXIM, B. R. Engenharia de Software: Uma abordagem Profissional. 8. ed. Porto Alegre: AMGH, 2016.

RYAN, R. M.; DECI, E. L. Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. American Psychologist. <https://doi.org/10.1037/0003-066X.55.1.68>.

SILVA, F. C. C. da. Visualização de dados: passado, presente e futuro. Liinc em Revista, [S.I.], v. 15, n. 2, 2019. DOI: 10.18617/liinc.v15i2.4812. Disponível em: <https://revista.ibict.br/liinc/article/view/4812>. Acesso em: 20 set. 2024.

SOMMERVILLE, I. Engenharia de Software. 9. ed. São Paulo: Pearson, 2013.

STATCOUNTER. Mobile Operating System Market Share Brazil. Disponível em: <<https://gs.statcounter.com/os-market-share/mobile/brazil>>. Acesso em: 1 ago. 2024.