

## Aplicativo UFSCar Planner

Adriano T. S. A. de Castro<sup>\*</sup>, Gabrelly C. Guimarães<sup>\*</sup>,  
Giullio E. da C. Di Gerolamo<sup>\*</sup>, Jorge L. M. Pires<sup>\*</sup>, Marcelina M. A. Maye<sup>\*</sup>,  
Matheus R. de Carvalho<sup>\*</sup>, Natália B. Magalhães<sup>\*</sup>, Renato Bueno<sup>†</sup>

Departamento de Computação  
Universidade Federal de São Carlos (UFSCar) - São Carlos, SP - Brasil

**Abstract.** *This article presents the development of UFSCar Planner, an application produced by the PET-BCC group at the Federal University of São Carlos through the agile Kanban methodology. It is aimed at the community's students and has the primary goal of facilitating the administration of academic life for its users through features such as managing events, consulting the University Restaurant menu, and monitoring attendance in subjects.*

**Resumo.** *Este artigo apresenta o desenvolvimento do UFSCar Planner, um aplicativo produzido pelo grupo PET-BCC da Universidade Federal de São Carlos por meio da metodologia ágil Kanban. Ele é voltado para os estudantes da comunidade e tem como objetivo primordial facilitar a administração da vida acadêmica de seus usuários através de funcionalidades como gerenciar eventos, visualizar o cardápio do Restaurante Universitário e controlar a frequência em disciplinas.*

### 1. Introdução

O avanço da tecnologia digital transformou consideravelmente a dinâmica entre as instituições de ensino superior e seus estudantes, influenciando tanto a interação quanto a administração das rotinas acadêmicas. O UFSCar *Planner* foi criado para atender às necessidades da comunidade acadêmica da Universidade Federal de São Carlos (UFSCar).

O UFSCar *Planner* tem origem no Laboratório de Inteligência Computacional e Engenharia (LINCE) do Departamento de Computação da UFSCar. O projeto foi desenvolvido pelos alunos com o apoio da empresa TokenLab sob a supervisão do Prof. Dr. Cesar Augusto Camilo Teixeira. No entanto, ao passar do tempo, a manutenção não foi possível devido à natureza fechada do código-fonte que foi desenvolvido em parceria com uma empresa privada, ou seja, novos estudantes não conseguiram continuar o desenvolvimento e manutenção do *Planner*. Além disso, o aplicativo teve o alcance limitado porque só estava disponível na plataforma APK Pure [APK Pure 2014].

Em 2021, os membros do grupo do Programa de Educação Tutorial do curso de Ciências da Computação (PET-BCC) lideraram a revitalização do UFSCar *Planner*, assumindo o projeto como uma atividade do grupo. Inicialmente a implementação estava prevista para ser feita em Flutter [Flutter 2015]. O processo teve início com a criação

---

<sup>\*</sup>Membro discente do Grupo do Programa de Educação Tutorial do curso de Ciências da Computação da Universidade Federal de São Carlos - PET-BCC UFSCar

<sup>†</sup>Tutor do Grupo do Programa de Educação Tutorial do curso de Ciências da Computação da Universidade Federal de São Carlos - PET-BCC UFSCar

de protótipos de telas, que foram adaptações da versão anterior do aplicativo. Ao longo desse período, a equipe decidiu migrar para o React Native [React Native 2015], fazendo uso da plataforma Expo [Expo 2015], que proporciona a capacidade de criar aplicativos compatíveis com diversas plataformas, incluindo o iOS [iOS 2007]. Embora fosse projetado para ser lançado a versão para iOS, apenas a versão para Android [Android 2021] foi lançada inicialmente.

Neste artigo será descrita a metodologia de desenvolvimento do aplicativo UFSCar *Planner*. Este artigo aborda as principais funcionalidades e as lições aprendidas com relação à escolha de tecnologias e às barreiras enfrentadas durante o processo de implementação. Além disso, é discutido o impacto e as perspectivas futuras deste aplicativo no contexto da comunidade acadêmica da Universidade Federal de São Carlos.

## 2. Metodologia

Para implementar a versão idealizada pelos membros do PET-BCC e com base no *Material Design 3* [Google 2021], foram realizadas reuniões de planejamento com frequência semanal. As tarefas de desenvolvimento foram divididas em dois grupos: prototipação e implementação. Nas reuniões de desenvolvimento eram debatidos os problemas encontrados, eventuais prioridades das tarefas a serem realizadas, assim como também ocorreu desenvolvimento em conjunto por parte do grupo.

As tarefas eram registradas e atribuídas utilizando o Método *KanBan* [Kanban 2009], o qual consiste na utilização de cartões para representar tarefas, organizados em colunas indicativas do estado de andamento da atividade (“A fazer”, “Fazendo” e “Feito”). Este método ágil foi escolhido por facilitar a visualização do progresso do projeto sem impor restrições rigorosas à sua prática, além de ser familiar aos colaboradores do projeto. Além disso, foram adicionadas novas categorias de cartões para suprir a necessidade de ter onde registrar novas ideias de funcionalidades para o aplicativo e erros encontrados durante os testes. O painel *KanBan* do projeto foi organizado na plataforma Trello [Trello 2011].

Ao término de cada tarefa de implementação foram feitos testes de regressão, cujo objetivo é verificar se alguma funcionalidade foi afetada negativamente pela modificação.

Embora o *Scrum* [Morais, I.S. (org.) 2017] tenha sido considerado para o projeto, constatou-se que as restrições de ter cerimônias como *daily*s, *planning*, *sprint review* não tinham espaço no projeto devido a falta de horários comuns entre os membros do grupo. Além disso, o fato do projeto não possuir um cliente real durante o desenvolvimento para receber as funcionalidades no final de cada *sprint* contribuiu para a não utilização do *Scrum*.

## 3. Prototipação

A prototipação, etapa crucial precedendo o desenvolvimento da aplicação, foi responsável por definir os recursos a serem incluídos no aplicativo, bem como a experiência do usuário e o *design* das telas, empregando técnicas de UX/UI (*User Experience/ User Interface*) [Norman 1988].

Para avaliar a usabilidade, adotou-se a técnica de *Benchmarking* [Levy 2015], com a realização de reuniões online com a equipe do projeto, nas quais aplicativos similares

funcionalidades desejadas pelos usuários. Essas atividades geraram documentos internos que orientaram as fases subsequentes do projeto.

A prototipação foi realizada por meio de reuniões online, utilizando o Figma [Figma 2016] para criar os *designs* e protótipos da aplicação. Os desenhos evoluíram de esboços simples para telas mais detalhadas, imitando uma aplicação real. Nestas etapas, a equipe avaliava a usabilidade e a acessibilidade de cada recurso proposto.

Inicialmente, o processo de *design* foi guiado pelo *Material Design 2* [Google 2018], padrão mantido principalmente pela Google [Google 1998] para orientar o desenvolvimento de aplicações móveis e web. No entanto, com o lançamento do *Material Design 3* [Google 2021] durante o período de prototipação, a equipe decidiu atualizar o aplicativo para esta nova versão, tornando o aplicativo mais moderno e acessível. No Figma, utilizou-se a funcionalidade de componentização para criar “*Design Tokens*”, conforme definido pela Google, consistindo em estilos tipográficos, cores e componentes reutilizáveis, assegurando uma consistência no *design* do protótipo, mesmo com a colaboração de vários integrantes.

Durante as iterações de *design*, foram realizados testes de usabilidade com estudantes que não faziam parte da equipe de desenvolvimento, técnica nomeada como “*Design Iterativo*” [Norman 1988]. O objetivo foi verificar a aceitação do aplicativo e identificar possíveis problemas de usabilidade que poderiam ter sido ignorados pela equipe. Os resultados desses testes foram documentados e utilizados para refinar o protótipo.

#### 4. Implementação

Com o auxílio de um painel compartilhado no Trello, foi aplicada a metodologia ágil *KanBan*, que, devido sua visualização facilitada do andamento geral do projeto e das tarefas a serem realizadas, direcionou a tomada de decisões referentes ao prosseguimento da implementação. Foram realizadas reuniões semanais para a verificação do progresso das tarefas, resolução de eventuais desafios encontrados na implementação e designação de novas tarefas.

Devido à introdução de uma nova tecnologia alheia aos conhecimentos prévios dos membros do grupo, a implementação progrediu em conjunto com o estudo do React Native. As tarefas designadas também seguiram a curva de aprendizagem, ou seja, inicialmente, as tarefas julgadas pelos membros do grupo como menos complexas receberam maior prioridade.

Nas seções a seguir são apresentadas algumas funcionalidades do aplicativo, juntamente com tecnologias específicas utilizadas em cada etapa e desafios de implementação.

##### 4.1. Cardápio do Restaurante Universitário

O acesso rápido às informações do cardápio do restaurante é uma das principais funcionalidades do aplicativo. Recuperar as informações sobre as refeições não foi uma tarefa trivial. Anteriormente a novembro de 2020, essas informações estavam disponíveis em formato tabular no site da UFSCar [UFSCar 2020], o qual o aplicativo do LINCE utilizava para obter informações. Posteriormente, as informações passaram a ser disponibilizadas em formato de imagens.

Para melhor organizar a obtenção dos dados, foi criada uma *API Restful* utilizando Django Rest Framework [Django Rest Framework 2005] a qual foi denominada RU API [PET-BCC 2020]. A primeira versão possuía a funcionalidade de obter as imagens, realizar um pré-processamento e utilizar a ferramenta *Open Source* de reconhecimento de caracteres Tesseract [Smith 2007] para extrair informações das imagens do cardápio. As imagens recuperadas com informação do cardápio mudavam de *template* constantemente, implicando no redesenvolvimento das técnicas de pré-processamento para se adaptar ao novo estilo de imagem. Sem o pré-processamento adequado não era possível utilizar o *tesseract* para recuperar totalmente a informação contida na imagem.

Assim como não era possível extrair facilmente as informações usando a API, também não era possível para pessoas com deficiência visual obter tais informações utilizando ferramentas assistivas. Em conjunto com Coordenadoria da Rede Integrada de Segurança Alimentar da UFSCar (CRISA), atual responsável pela publicação do cardápio, foi debatida a questão do uso e acessibilidade dos dados, cunhando um acordo para disponibilização do cardápio em texto plano no site, tanto para o aplicativo quanto para as ferramentas assistivas, resultando na atual versão da RU API, utilizada na versão atual do aplicativo UFSCar *Planner*.

#### **4.2. Gerenciamento de estados e configurações do usuário**

O gerenciamento de estados visa manter o usuário devidamente informado sobre as atividades de relevância que ocorrem na UFSCar, com o objetivo de facilitar a organização das tarefas dos estudantes.

Cada atividade registrada no aplicativo disponibiliza informações essenciais, a saber: a especificação da atividade, o local onde será realizada, a data e o horário previstos para realização da mesma. No caso de atividades acadêmicas, é apresentado o nome do ministrante. Já para as palestras, é exibido o nome do palestrante.

As atividades são categorizadas em três diferentes estados: “marcadas”, que corresponde às atividades que o usuário agendou previamente; “acontecendo agora”, referente às atividades que estão ocorrendo em tempo real; e “concluídas”, que engloba as atividades que já foram realizadas ou para as quais o prazo estipulado expirou.

Além disso, na aba “configurações” o usuário pode personalizar o arranjo do aplicativo de acordo com as suas preferências. Esta personalização pode abranger a modificação do tema visual da interface, a inserção do valor correspondente a uma refeição, a definição do período acadêmico, especificando a data de início e término, bem como a escolha do nome pelo qual deseja ser identificado.

#### **4.3. Exibição de disciplinas**

A funcionalidade “Exibição de Disciplinas” permite que o usuário visualize na aba “*Planner*” a sua agenda de aulas para a semana, como pode ser visto na Figura 1, em que é apresentada a interface do aplicativo (com exemplos dos temas claro e escuro). Além de consultar os horários, é possível clicar no botão em que é exibido o local da disciplina e ser redirecionado ao Google Maps [Google 2005] no endereço da aula. Permite também adicionar notas de atividades e cálculo da média de cada disciplina, bem como registrar a frequência, proporcionando ao estudante maior controle sobre a sua vida acadêmica.

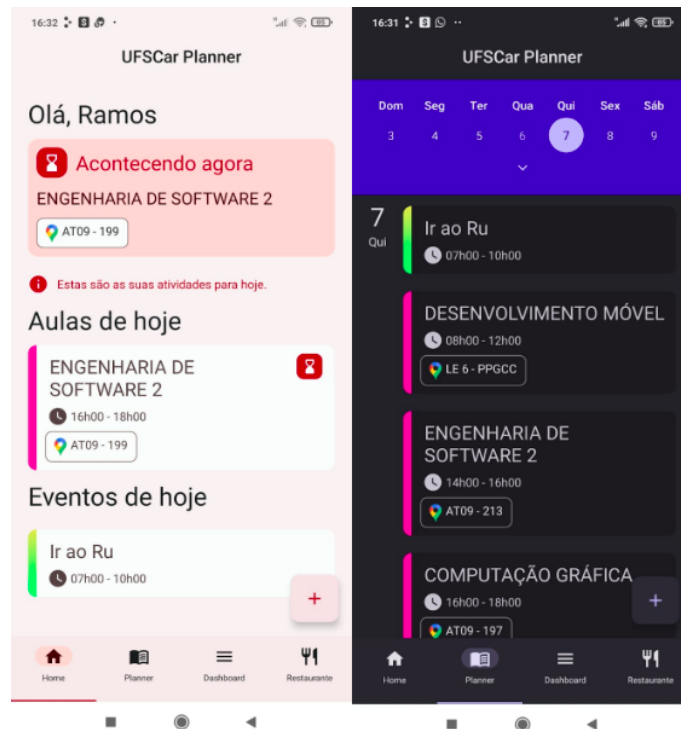


Figura 1. Aba “Planner” do UFSCar Planner em tema claro e escuro.

O aplicativo utiliza a API do Sistema Integrado de Gestão Administrativa (SIGA) da universidade para obter as informações sobre as disciplinas do usuário. Esta API foi cedida aos desenvolvedores e é mantida pela Secretaria Geral de Informática (SIn) - órgão gestor e provedor de soluções de tecnologia da informação da UFSCar - e possui acesso público. Portanto, através do login acadêmico de um estudante, obtém-se suas informações no formato JSON [JSON 2000] via API, que são tratadas e exibidas para o usuário.

#### 4.4. Saldo da Carteirinha Universitária

Já durante o processo de desenvolvimento do UFSCar *Planner*, uma nova funcionalidade foi introduzida, que permite aos usuários monitorar o saldo da sua carteirinha do Restaurante Universitário (RU) e o número de refeições disponíveis. Essa adição foi diretamente inspirada na funcionalidade disponibilizada no site SAGUI UFSCar [UFSCar 2021], com o objetivo de proporcionar maior conveniência e controle aos usuários.

Vale destacar que essa funcionalidade pode ser utilizada de duas maneiras distintas, dependendo da preferência do usuário. Primeiramente, quando o aplicativo é acessado, solicita-se permissão ao usuário para coletar o número de RA/CPF e senha cadastrada previamente no sistema UFSCar. Caso o usuário opte por negar essa permissão, a funcionalidade passa a operar de maneira manual. Isso significa que o usuário terá que realizar ajustes no saldo remanescente da carteirinha ou debitar refeições manualmente, sem que haja intermediação do sistema. Por outro lado, se o usuário conceder a permissão solicitada e inserir corretamente suas informações, a funcionalidade irá operar de forma automática. Nesse cenário, o aplicativo atualiza o saldo da carteirinha e o número de refeições disponíveis sempre que é iniciado, consultando diretamente o servidor do

SIGA. Essa abordagem simplifica o processo e oferece aos usuários um acompanhamento automatizado e preciso de suas contas no RU.

Inicialmente, foi implementada apenas a função manual no aplicativo, permitindo ao usuário debitar refeições e ajustar o valor da refeição individual, além de alterar o saldo total. Somente posteriormente, introduziu-se a capacidade de automatizar essa funcionalidade, utilizando a API do site SAGUI UFSCar. No entanto, em razão da necessidade de coletar dados para monitorar o saldo em tempo real, foi preciso realizar uma atualização na política de privacidade vinculada a esse aspecto.

## 5. Resultados

Nesta seção, serão discutidos os resultados obtidos com a implementação do aplicativo, a partir dos dados retirados diretamente da Google Play Store [Google 2012]. Hoje, o aplicativo conta com mais de dois mil e duzentos usuários, sendo que alguns destes decidem compartilhar seus dados com a plataforma da Google.

Na Figura 2 é possível analisar as entradas de novos usuários para o aplicativo UFSCar *Planner* em um período de 30 dias. Observa-se um aumento significativo de usuários adquiridos no começo do mês de outubro de 2023, o qual deve-se à volta as aulas dos estudantes da graduação, impulsionado também pela maior divulgação do aplicativo. Com a análise desse tipo de dados pode-se evidenciar uma constante rotatividade de usuários, considerando o ingresso de novos estudantes ou a descoberta do aplicativo durante a graduação, enquanto outros concluem seus cursos e abandonam o uso do aplicativo.



**Figura 2. Aquisição e Perda de usuários durante um período de 30 dias.**

Na Figura 3 são apresentados os erros detectados no aplicativo no mesmo período de 30 dias. É visível que com o aumento de usuários os erros detectados também aumentaram. Esse aumento está completamente associado a significativa ampliação de *downloads* e compartilhamento de dados pelos novos usuários. Verifica-se que para uma média de dois mil e duzentos usuários, uma taxa de 30 erros por mês consolida uma certa estabilidade no aplicativo. Além disso, a falta de erros em todo o mês de setembro se deve a falta de utilização da plataforma durante as férias dos estudantes no período citado.



**Figura 3. Erros durante um período de 30 dias.**

Com a ajuda de todos esses dados coletados pela Google, se torna aparente uma evolução do uso do aplicativo em início de semestres, mantendo uma regularidade durante o período de aulas. Também vê-se o fenômeno oposto no período das férias, onde os estudantes deixam de usar o aplicativo, além daqueles que se formam.

## 6. Conclusão

Este artigo aborda a criação do aplicativo UFSCar *Planner*, que foi projetado para atender às necessidades dos estudantes da Universidade Federal de São Carlos (UFSCar) em relação à gestão de suas vidas acadêmicas. Este aplicativo passou por várias mudanças e desafios ao longo dos anos, até chegar à versão mais recente, de agosto de 2023.

A prototipação de várias versões do aplicativo que mostram disciplinas, cardápio do restaurante universitário e informações sobre eventos acadêmicos foram parte do processo de desenvolvimento. A migração para a plataforma React Native e o uso do Expo para o desenvolvimento de aplicativos multiplataforma aumentaram a acessibilidade e a flexibilidade do UFSCar *Planner*.

Além disso, o artigo abordou os problemas associados à aquisição do cardápio do Restaurante Universitário, enfatizando que a acessibilidade e a cooperação com a CRISA são essenciais para garantir que as informações sejam acessíveis e disponíveis. Para fornecer uma experiência completa e personalizada aos usuários, outras funcionalidades importantes, como gerenciamento de estado e configurações do usuário, exibição de disciplinas e saldo da carteirinha universitária, foram adicionadas.

O aplicativo UFSCar *Planner* contribuiu para a melhoria na gestão e organização da vida acadêmica dos estudantes da UFSCar. Este aplicativo não apenas ajuda os estudantes a organizar suas atividades acadêmicas, mas também oferece recursos adicionais que facilitam a rotina universitária.

## Referências

- Android (2021). Android. [https://www.android.com/intl/pt-BR\\_br/](https://www.android.com/intl/pt-BR_br/). [Acesso em: 26/09/2023].
- APK Pure (2014). Plataforma APK Pure. <https://apkpure.com/br/>. [Acesso em: 26/09/2023].

- Django Rest Framework (2005). Django Rest Framework. <https://www.django-rest-framework.org/>. [Acesso em: 26/09/2023].
- Expo (2015). Expo. <https://expo.dev>. [Acesso em: 26/09/2023].
- Figma (2016). Figma. <https://figma.com>. [Acesso em: 26/09/2023].
- Flutter (2015). Flutter. <https://flutter.dev>. [Acesso em: 26/09/2023].
- Google (1998). Google. <https://www.google.com.br/>. [Acesso em: 14/11/2023].
- Google (2005). Google maps. <https://www.google.com/intl/pt-br/maps/about/#!/>. [Acesso em: 26/09/2023].
- Google (2012). Google play store. <https://play.google.com/>. [Acesso em: 26/09/2023].
- Google (2018). Material Design 2. <https://m2.material.io>. [Acesso em: 26/09/2023].
- Google (2021). Material Design 3. <https://m3.material.io>. [Acesso em: 26/09/2023].
- iOS (2007). ios. <https://www.apple.com/br/ios/ios-17/>. [Acesso em: 26/09/2023].
- JSON (2000). Introdução ao JSON. <https://www.json.org/json-pt.html>. Acesso em: 26/09/2023.
- Kanban (2009). Kanban. <https://www.atlassian.com/agile/kanban#:~:text=What%20is%20kanban%3F,of%20work%20at%20any%20time>. [Acesso em: 26/09/2023].
- Levy, J. (2015). *UX Strategy*. O'Reilly Media.
- Morais, I.S. (org.) (2017). *Engenharia de Software*. Pearson, São Paulo, 1 edition. E-book.
- Norman, D. (1988). *Design do dia a dia*. Anfiteatro.
- PET-BCC (2020). Repositório do ru api. [https://github.com/petbccufscar/ru\\_api](https://github.com/petbccufscar/ru_api). [Acesso em: 26/09/2023].
- React Native (2015). React Native. <https://reactnative.dev>. [Acesso em: 26/09/2023].
- Smith, R. (2007). An overview of the tesseract ocr engine. In *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, volume 2, pages 629–633.
- Trello (2011). Trello. <https://trello.com>. [Acesso em: 26/09/2023].
- UFSCar (2020). Aba de restaurante do site da ufscar. <https://www.ufscar.br/restaurantes-universitario/cardapio>. [Acesso em: 26/09/2023].
- UFSCar (2021). SAGUI UFSCar. <https://sistemas.ufscar.br/sagui/>. [Acesso em: 14/11/2023].