

Universidade Federal do Tocantins

Projeto de sistemas

Aplicativo Moovin

Data 12/05/2025

Alunos: Victhor C. Magalhães, João Vitor Reis Dias, Mayconn Cardoso Soares, Pedro Lucas, Ana Julia

## Planejamento da 3º Sprint

**Sprint 3:** A Sprint 3 do projeto Moovin tem como objetivo implementar funcionalidades essenciais para melhorar a experiência dos usuários e a eficiência do sistema. Entre as principais entregas estão a implementação de um sistema de notificações para manter os usuários informados sobre eventos importantes, como novas mensagens e avaliações recebidas, além do acompanhamento de prazos de aluguel para garantir que os usuários possam gerenciar suas locações de forma eficiente. Outra funcionalidade importante é o agendamento de visitas a imóveis, voltado para usuários premium, facilitando a comunicação e interação com os proprietários. Além disso, será desenvolvido o sistema de plano premium para inquilinos e proprietários, proporcionando acesso a recursos exclusivos, e a refatoração das permissões do backend para garantir um controle de acesso mais seguro e organizado. Também será realizada a refatoração dos menus de navegação e telas do aplicativo, garantindo uma navegação intuitiva e uma interface mais amigável. Essas melhorias visam otimizar a usabilidade, aumentar o engajamento dos usuários e oferecer novas oportunidades de monetização para o aplicativo.

**Valor da sprint:** A Sprint 3 do projeto Moovin tem como objetivo implementar funcionalidades-chave que vão proporcionar um valor significativo tanto para os usuários quanto para a estrutura do sistema. As melhorias incluem sistemas de notificações, agendamento de visitas, acompanhamento de prazos de aluguel, planos premium, refatoração das permissões e melhorias na navegação. Essas implementações têm impacto direto na usabilidade, na experiência do usuário e na segurança, além de trazerem oportunidades de monetização e fidelização.

### **Cronograma:**

Início: 12/05

Primeira Revisão: 20/05

Segunda Revisão: 24/05

Entrega: 26/05

## **Requisitos da sprint:**

### **1. Sistema de Notificações**

**Responsável:** Mayconn

**Revisores:** Pedro e Ana Júlia

#### **Funcionalidades:**

- **Backend:**
  - Criação de uma tabela no banco de dados para armazenar notificações (ID, tipo de notificação, mensagem, status de leitura, data/hora).
  - API para envio de notificações (tipo de evento: nova mensagem, nova avaliação, etc.).
  - Sistema de envio de notificações por push (notificações em tempo real).
  - Endpoint para marcação de notificações como lidas.
  - Notificações em tempo real via WebSocket ou outras tecnologias de push.
- **Frontend:**
  - Exibição das notificações em uma tela dedicada ou através de ícones de notificação no menu.
  - Marcação visual de notificações não lidas.
  - Possibilidade de acessar as notificações para mais detalhes (exemplo: ver a mensagem ou avaliação recebida).
  - Tela de configuração para gerenciar as preferências de notificação (habilitar/desabilitar notificações de mensagens, avaliações, etc.).

### **2. Acompanhamento de Prazos de Aluguel**

**Responsável:** João Vitor

**Revisores:** Victhor e Mayconn

#### **Funcionalidades:**

- **Backend:**
  - Tabela para armazenar informações sobre aluguéis, incluindo datas de início e término, status e ID do inquilino/proprietário. (Já feito)
  - API para consulta de prazos de aluguel com base no usuário logado.
  - Sistema de alertas ou notificações para lembrar o usuário sobre prazos de aluguel (exemplo: 3 dias antes do término do aluguel).
  - Endpoint para o usuário poder visualizar todos os alugueis e seus respectivos prazos.
- **Frontend:**
  - Tela para o usuário visualizar todos os alugueis ativos, com informações sobre os prazos e status.
  - Alertas ou notificações para lembrar os usuários sobre a data de término de aluguel.
  - Opção para os usuários marcarem a data do fim do aluguel, renovação ou outro status (se aplicável).
  - Interface amigável para facilitar a visualização de alugueis futuros e passados.

### 3. Agendamento de Visitas a Imóveis

**Responsável:** Pedro Lucas

**Revisores:** João Vitor e Ana Júlia

#### **Funcionalidades:**

- **Backend:**
  - Tabela para armazenar solicitações de visitas a imóveis com informações sobre o imóvel, data e hora desejadas para a visita, status (pendente, confirmada, cancelada).
  - API para agendamento de visitas e acompanhamento de status (criação, cancelamento, confirmação).
  - Validação para garantir que as visitas sejam feitas em horários disponíveis.
- **Frontend:**
  - Interface onde o usuário premium pode visualizar detalhes do imóvel e agendar visitas.
  - Calendário interativo ou seleção de data e hora para agendamento de visitas.
  - Tela de histórico de visitas, com status de cada solicitação.
  - Opção de confirmação ou cancelamento das visitas agendadas.

- Integração com o backend para o agendamento e atualização do status.

## 4. Plano Premium para Inquilinos e Proprietários

**Responsável:** Victhor

**Revisores:** Mayconn

**Funcionalidades:**

- **Backend:**
  - Sistema de controle de planos premium para inquilinos e proprietários (com tabelas específicas para gerenciar assinaturas, pagamentos, renovação de planos).
  - API para verificar o status do plano premium de um usuário (ativo ou expiring).
  - Controle de acesso a funcionalidades específicas para usuários com plano premium (exemplo: agendamento de visitas, acesso a imóveis exclusivos).
- **Frontend:**
  - Tela dedicada ao gerenciamento do plano premium, com opções de visualização e compra do plano.
  - Exibição de informações do plano atual (datas de validade, benefícios, etc.).
  - Botões para ativação do plano premium e seus benefícios.
  - Notificações ou alertas para informar ao usuário sobre a expiração ou renovação do plano.

## 5. Refatorar Permissões do Backend

**Responsável:** Victhor

**Revisores:** Pedro

**Funcionalidades:**

- **Backend:**

- Refatoração do sistema de permissões no backend para garantir que diferentes tipos de usuários (inquilinos, proprietários, administradores) tenham acessos adequados a recursos específicos.
- API para validação de permissões ao realizar operações específicas (exemplo: edição de perfil de imóvel para proprietário, acesso a detalhes de aluguel para inquilinos).
- Criação de um sistema de roles (funções) para categorizar as permissões de cada tipo de usuário.
- **Frontend:**
  - Garantir que o frontend lide corretamente com os diferentes níveis de acesso. Por exemplo:
    - Tela de perfil do inquilino deve mostrar apenas informações relevantes para o inquilino.
    - Tela de gerenciamento de imóveis deve ser acessível apenas para o proprietário.
  - Feedback visual claro para o usuário quando ele não tiver permissões para acessar determinada funcionalidade.

## 6. Menus de Navegação e Refatoração de Telas

**Responsável:** Ana Júlia

**Revisores:** João Vitor e Victhor

**Funcionalidades:**

- **Frontend:**
  - Refatoração das telas para garantir que o fluxo de navegação entre elas seja intuitivo e fácil de usar.
  - Criação de menus de navegação claros, com categorias e ícones que orientem o usuário.
  - Implementação de um sistema de navegação baseado em BottomNavigationBar, Drawer ou outros componentes, dependendo do design.
  - Integração com o backend para exibição de dados dinâmicos (por exemplo, dados do perfil, imóveis favoritos, etc.).
- **Backend:**
  - Garantir que as APIs estejam configuradas corretamente para fornecer os dados necessários para as telas de navegação.

- Integração de autenticação e permissões para garantir que o usuário tenha acesso às páginas corretas.