# Relatório FinancePilot – Consolidado por Semanas

Desenvolvimento do Projeto em Angular – Julho de 2025

## Semana 1

Atividades Realizadas – Todas as atividades da primeira semana foram elaboradas em conjunto.  
  
- Foram elaboradas 7 propostas iniciais de projetos, com foco em inovação e impacto no dia a dia dos usuários.  
- Escolhemos uma proposta principal para dar continuidade: o FinPilot, um copiloto financeiro voltado à gestão financeira de micro e pequenas empresas, além de profissionais autônomos.  
- Refinamos a ideia central do projeto, definindo seu objetivo, diferenciais e funcionalidades prioritárias.  
- Estruturamos o escopo inicial e mapeamos 7 telas principais para a primeira versão do sistema.  
- Criamos o repositório no GitHub e organizamos o ambiente de desenvolvimento colaborativo.

## Semana 2

05/07/2025 - Sábado  
- Prototipar Login e Cadastro – Josias  
- Prototipar Landing page e começar implementação das interfaces – Lucas  
- Início da lista de requisitos – Josias  
  
06/07/2025 - Segunda-feira  
- Prototipada interface de administração – Lucas  
  
08/07/2025 - Quarta e Quinta-feira  
- Ajustes nas interfaces e padronização de elementos – Lucas

🔹 06 a 10 de Julho de 2025

- Tela de Administração

- Implementação da interface administrativa com foco em gestão de usuários. – Lucas.

- Construção de componentes independentes para exibição, edição e exclusão de registros. (Lucas)

- Estrutura modular e organizada, seguindo boas práticas com Angular Standalone. - Lucas.

- Inclusão da rota de administração no sistema de navegação. - Lucas

- Estilização visual com SCSS alinhado ao restante do projeto. - Lucas

- Preparação da tela para futura integração com autenticação e controle de permissões. - Lucas

## Semana 3

Atividades realizadas entre 13 e 25 de Julho de 2025 no desenvolvimento front-end com Angular:

### Resumo das Atividades

🔹 13 a 15 de Julho  
- Tela de Login e Cadastro melhoria de visual e componentes - Josias  
 - Criação das telas de autenticação (login e cadastro). - Lucas  
 - Uso de Reactive Forms com validações básicas. - Josias  
 - Integração visual com Angular Material (MatFormField, MatInput, MatButton, MatIcon). - - Josias  
 - Estilização escura aplicada com SCSS. - Josias  
  
🔹 16 de Julho  
- Tela de Cadastro – Lucas  
 - Desenvolvimento com foco visual e temático (finanças e aviação). - Josias  
 - Inclusão de ícones (MatIcon) e ilustrações 3D. - Josias  
 - Formulário com Reactive Forms, validações e design responsivo. - Josias  
  
🔹 17 a 20 de Julho  
- Tela de Pedidos - Josias  
 - Filtros reativos: Período, Método de Pagamento, Status do Repasse. - Josias  
 - MatSelect para filtros com integração ao formGroup. - Josias  
 - Estilização de status com cores distintas (.status-pago, .status-pendente). - Josias  
 - Inclusão de checkbox individual e global para seleção múltipla. - Josias  
 - Botões de edição e remoção por pedido. - Josias  
  
🔹 21 a 23 de Julho  
- Tela de Receitas e Despesas - Josias  
 - Filtros inteligentes por tipo, período e categoria. - Josias  
 - Criação de cards de resumo: Receitas, Despesas, Saldo. - Josias  
 - Tabela de lançamentos com ações e status visuais. - Josias  
 - Layout fluido com CSS Grid. - Josias  
  
🔹 24 a 25 de Julho  
- Refatorações e Melhorias Técnicas - Lucas  
 - Correção de erros NG8002 e NG0400. - Lucas  
 - Refinamento visual: bordas, espaçamentos e cores. - Josias  
 - Ajuste do componente app-navbar. - Josias  
 - Reorganização dos componentes standalone. - Lucas

### Observação Importante

Até o momento, os dados utilizados nas tabelas e cards ainda não foram mockados com services.  
  
Justificativa:  
A decisão de adiar a criação dos dados mockados foi tomada para priorizar a construção e validação da estrutura visual e funcional das telas. Como o layout, navegação e filtros exigiram ajustes finos com base no Figma, optou-se por primeiro garantir a consistência visual e comportamental do front-end antes de injetar dados simulados.  
  
A mockagem dos dados será incluída na próxima fase, com uso de Observable, BehaviorSubject ou HttpClient simulando APIs locais.

### Tecnologias & Bibliotecas Utilizadas

- Angular 17+ com Standalone Components  
- Angular Material (FormField, Input, Select, Checkbox, Button, Icon)  
- ReactiveFormsModule e FormsModule  
- SCSS Modular  
- Componentização com Navbar, Sidebar, CardResumo, TabelaPedidos, etc.