= SPEC-001: Aplicativo Django de Controle Financeiro

:sectnums:

:toc:

== Background

A necessidade de um controle financeiro pessoal eficiente e acessível é cada vez mais premente em um mundo onde as despesas são variadas e complexas. O aplicativo de controle financeiro visa fornecer uma ferramenta intuitiva e abrangente para que os usuários possam gerenciar suas finanças pessoais, acompanhar despesas, visualizar relatórios financeiros detalhados, e planejar gastos futuros com maior precisão.

== Requirements

\*Controle de Despesas\*: Permitir que os usuários registrem suas despesas diárias, categorizando-as para uma análise mais detalhada.

\*Geração de Relatórios Financeiros\*: Fornecer relatórios detalhados sobre o estado financeiro do usuário, incluindo gastos por categoria, evolução das despesas ao longo do tempo, etc.

\*Dashboards\*: Apresentar uma visão geral rápida das finanças do usuário através de dashboards intuitivos, mostrando gastos recentes, tendências de despesas, e alertas de orçamento.

\*Controle de Gastos Futuros\*: Permitir que os usuários planejem seus gastos futuros, criando orçamentos para categorias específicas de despesas.

Todos os usuários terão acesso comum ao sistema, sem diferenciação de permissões.

== Method

A arquitetura do aplicativo seguirá um modelo tradicional do Django, com uma separação clara entre as camadas de Modelo, Visão e Template (MVT), integrando a última versão do Bootstrap para os templates do frontend.

=== Arquitetura do Aplicativo

- \*Modelos\*: Usuário (com email para login), Despesa, Categoria (e Subcategoria), Orçamento, Cartão de Crédito, Cheque.

- \*Views\*: Para controle de despesas, geração de relatórios, visualização de dashboards, planejamento de orçamentos, gestão de cartões de crédito, e acompanhamento de cheques.

- \*Templates\*: Utilização do Bootstrap para design responsivo.

=== Esquema de Banco de Dados

- \*Usuário\*: Informações dos usuários, usando email como identificador para login.

- \*Despesa\*: Registros de despesas, com valor, data, subcategoria, e usuário.

- \*Categoria e Subcategoria\*: Para classificação detalhada das despesas.

- \*Orçamento\*: Para planejamento financeiro dos usuários.

- \*Cartão de Crédito\*: Gestão de informações do cartão de crédito.

- \*Cheque\*: Controle de cheques emitidos.

=== Considerações de Escalabilidade

O sistema será projetado para ser escalável, permitindo fácil expansão e ajuste a um volume crescente de usuários e transações.

== Implementation

O desenvolvimento do aplicativo será estruturado em fases, cada uma focada em um conjunto específico de funcionalidades. A implementação seguirá as práticas ágeis, permitindo iterações rápidas e feedback contínuo.

1. Configuração do Ambiente de Desenvolvimento:

- Configurar um ambiente virtual Python.

- Instalar Django e outras dependências necessárias.

- Configurar o projeto Django e aplicativos individuais para controle de despesas, relatórios, etc.

2. Implementação do Modelo de Dados:

- Definir modelos para Usuário, Despesa, Categoria, Subcategoria, Orçamento, Cartão de Crédito, e Cheque no Django.

- Implementar migrações para criar o esquema de banco de dados.

3. Desenvolvimento das Views e Templates:

- Implementar as views para manipular a lógica de negócios das funcionalidades principais.

- Criar templates usando Bootstrap para as interfaces do usuário.

4. Funcionalidades de Autenticação:

- Configurar o sistema de autenticação do Django para usar o email como identificador de login.

- Implementar funcionalidades de registro, login, e recuperação de senha.

5. Implementação de Testes:

- Escrever testes unitários e de integração para assegurar a qualidade e a funcionalidade do código.

6. Preparação para Produção e Lançamento:

- Configurar um servidor de produção adequado.

- Realizar deploy do aplicativo.

- Monitorar e otimizar o desempenho do aplicativo.

== Milestones

1. \*\*Configuração Inicial e Estrutura do Projeto\*\* - Conclusão da configuração do ambiente e estrutura básica do projeto.

2. \*\*Modelo de Dados Completo\*\* - Modelos definidos e esquema de banco de dados implementado.

3. \*\*Primeira Versão Funcional\*\* - Implementação básica de funcionalidades, autenticação de usuário, e interface do usuário.

4. \*\*Cobertura de Testes Alcançada\*\* - Cobertura de testes significativa, assegurando a estabilidade das funcionalidades implementadas.

5. \*\*Lançamento do MVP\*\* - Lançamento de uma versão mínima viável, incluindo todas as funcionalidades essenciais.

6. \*\*Otimização e Escalabilidade\*\* - Ajustes finais para otimização de desempenho e preparação para escalabilidade futura.

= SPEC-001: Aplicativo Django de Controle Financeiro

:sectnums:

:toc:

== Background

Este aplicativo visa fornecer uma ferramenta intuitiva e abrangente para gestão financeira pessoal, permitindo aos usuários gerenciar despesas, visualizar relatórios financeiros, e planejar gastos futuros.

== Requirements

Os requisitos incluem controle de despesas, geração de relatórios financeiros, dashboards, controle de gastos futuros, todos acessíveis por usuários comuns.

== Method

A solução será desenvolvida com Django, seguindo o padrão MVT, e incluirá modelos detalhados para usuários, despesas, categorias, subcategorias, orçamentos, cartões de crédito, e cheques. A escalabilidade será uma consideração chave, assim como a integração da última versão do Bootstrap para os templates do frontend.

== Implementation

O desenvolvimento seguirá uma abordagem ágil, dividido em fases que cobrem desde a configuração do ambiente, desenvolvimento das funcionalidades principais, até a preparação para o lançamento do produto.

== Milestones

Os marcos incluem a configuração inicial, modelo de dados completo, primeira versão funcional, cobertura de testes, lançamento do MVP, e otimização para escalabilidade.