



Documentação de Projeto

Projeto Integrador 3A (PDM-PI)

Disciplina:	Projeto Integrador 3A
Professor:	Bruno Miranda
Tipo de atividade:	Avaliativa - Projeto
Turma:	COD-Turma
Semestre	1º semestre de 2024
Departamento / Curso:	Ciência da Computação
Membros da equipe:	Aaron Willian (2122130037), Christian Soares (2122130002) ,Lucas Mendonça (2122130042)

SUMÁRIO

1. Projeto Integrador 3 2. Definições Iniciais do Projeto 3 Tema 3 Segmento 3 Público Alvo 3
3. Macro Escopo da Solução 3 4. Funcionalidades a serem Implementadas 3 5. Requisitos
Mapeados 4 Requisitos de usuário 4 Requisitos de negócio 4 Requisitos funcionais 4 6.
Matriz de Rastreabilidade 4 7. Descrição dos Casos de Uso 4 *Caso de uso 1* 4 *Caso de uso*
2 5 *Caso de uso* 3 5 8. Protótipos de Tela 5 9. Decisões de Arquitetura 5 10. Decisões de
Configuração 6 11. Ferramentas e Recursos Necessários 6 12. Referências 6

1. Projeto Integrador

O desenvolvimento de aplicativos móveis está se tornando cada vez mais essencial na área da Tecnologia da Informação, que proporciona soluções práticas para as necessidades do dia a dia dos usuários. Pensando nisso, este artigo demonstra o desenvolvimento de um aplicativo mobile chamado List Creator App (LCA), feito para a criação e gerenciamento de listas no contexto geral de forma intuitiva e eficiente. O LCA oferece uma interface simples e intuitiva, permitindo que os usuários criem, editem e compartilhem suas listas, bem como marcar itens como concluídos e criar listas de forma hierárquica. Para o desenvolvimento do LCA foi utilizado, Java, Android Studio dentre outras ferramentas. O aplicativo foi desenvolvido para facilitar a vida do usuário que se mantém organizado, ou precisa fazer compras em algum lugar ou até para criar uma rotina, são infinitas possibilidades. Este trabalho contribui para a área de Tecnologias Mobile ao apresentar uma solução prática e funcional para a criação e organização de listas, facilitando a vida cotidiana dos usuários.

2. Definições Iniciais do Projeto

Tema

- O tema escolhido para o projeto, foi um aplicativo de criação de listas, com uma fácil interação para o usuário, e possibilidade de editar da forma que quiser. Foi decidido este tema onde não limita a capacidade do usuário em criar um único tipo de lista mas sim infinitas possibilidades.

Público Alvo

Como foi informado, o público alvo não é bem específico, pois o aplicativo não se limita a um tipo de usuário mas sim para todos.

Exemplos:

- Usuários Organizados
- Profissionais de qualquer área
- Estudantes
- Pais e cuidadores
- Entusiastas da produtividade

3. Macro Escopo da Solução

Lista de funcionalidades mapeadas:

Id	Nome	Descrição	Complexidade percebida
1	<i>Criação de lista</i>	<i>Os usuários podem criar novas listas, atribuindo um nome e uma descrição a cada uma.</i>	<i>Baixa</i>
2	<i>Edição de Listas</i>	<i>As listas podem ser editadas a qualquer momento, permitindo a adição, remoção e alteração da lista.</i>	<i>Média</i>
3	<i>Compartilhamento</i>	<i>Os usuários podem compartilhar suas listas com outros usuários via e-mail e redes sociais.</i>	<i>Alta</i>
4	<i>Exclusão de Listas</i>	<i>Os usuários podem excluir listas que não são mais necessárias.</i>	<i>Baixa</i>
5	<i>Marcar Itens como Concluídos</i>	<i>Os usuários podem marcar itens nas listas como concluídos, facilitando o acompanhamento de tarefas.</i>	<i>Baixa</i>

4. Funcionalidades a serem Implementadas

Lista de funcionalidades que serão implementadas no protótipo:

Id	Nome Funcionalidade	Nome Responsável	Status
1	<i>Projeto Integrador</i>	<i>Lucas e Aaron Willian</i>	<i>Em Andamento</i>
2	<i>Definições Iniciais do Projeto</i>	<i>Cristhian, Lucas e Aaron Willian</i>	<i>Concluído</i>
2.1	<i>Tema</i>	<i>Christian e Lucas</i>	<i>Concluído</i>
2.2	<i>Segmento</i>	<i>Cristhian, Lucas e Aaron Willian</i>	<i>Concluído</i>
2.3	<i>Público alvo</i>	<i>Christian e Lucas</i>	<i>Concluído</i>
3	<i>Macro Escopo da Solução</i>	<i>Aaron Willian</i>	<i>Concluído</i>
4	<i>Requisitos mapeados</i>	<i>Christhian , Aaron Willian</i>	<i>Concluído</i>
4.1	<i>Requisitos de usuário</i>	<i>Aaron Willian</i>	<i>Concluído</i>

4.2	<i>Requisitos de negócios</i>	<i>Aaron Willian</i>	<i>Concluído</i>
4.3	<i>Requisitos funcionais</i>	<i>Christian</i>	<i>Concluído</i>

5. Requisitos Mapeados

Requisitos de usuário

1. Interface Intuitiva: Os usuários esperam uma interface simples e fácil de usar, que permita a criação, edição e compartilhamento de listas de forma intuitiva, sem a necessidade de instruções complicadas.
2. Funcionalidades Versáteis: Os usuários desejam poder criar listas de diferentes tipos, como listas de compras, listas de tarefas e listas de rotinas, além de poder marcar itens como concluídos e organizar listas hierarquicamente.
3. Facilidade de Compartilhamento: A capacidade de compartilhar listas com outros usuários é essencial. Isso pode incluir a capacidade de compartilhar listas completas ou itens individuais dentro de uma lista, facilitando a colaboração e a coordenação entre usuários.
4. Sincronização em Tempo Real: Os usuários esperam que as atualizações feitas em suas listas sejam refletidas instantaneamente em todos os dispositivos nos quais o aplicativo está instalado. Isso garante consistência e evita conflitos de dados.
5. Personalização: Os usuários desejam poder personalizar suas listas com cores, ícones ou etiquetas, permitindo uma organização visualmente atraente e personalizada.

Requisitos de negócio

1. Aumento da Produtividade: O objetivo principal do List Creator App (LCA) é aumentar a produtividade dos usuários, fornecendo uma solução eficiente para a criação e organização de listas, reduzindo o tempo gasto em tarefas administrativas.
2. Atingir um Amplo Mercado: O aplicativo visa atingir um amplo mercado de usuários que buscam uma solução simples e eficaz para suas necessidades de listas, desde profissionais que precisam gerenciar tarefas até pessoas que desejam organizar suas compras e rotinas diárias.
3. Manutenção e Suporte: É essencial fornecer manutenção contínua e suporte ao aplicativo para garantir sua estabilidade e segurança, além de responder às solicitações de suporte dos usuários de forma eficiente e diária.

Requisitos funcionais

1. Registro/Login de Usuário: O aplicativo deve permitir que os usuários criem uma conta ou faça login usando suas credenciais existentes.
2. Criação e Edição de Listas: Os usuários devem criar novas listas e editar

listas existentes, adicionando, removendo ou modificando itens conforme necessário.

3. Compartilhamento de Listas: O aplicativo deve permitir que os usuários compartilhem suas listas com outros usuários através de várias plataformas, como e-mail, mensagens de texto ou redes sociais.
4. Marcação de Itens: Os usuários devem poder marcar itens como concluídos em suas listas, bem como desfazer ou refazer ações.
5. Organização Hierárquica: O aplicativo deve suportar a organização hierárquica de listas, permitindo a criação de listas principais e sub listas aninhadas.
6. Sincronização em Tempo Real: As atualizações feitas em uma lista em um dispositivo devem ser refletidas instantaneamente em todos os outros dispositivos nos quais o aplicativo está instalado.
7. Personalização de Listas: Os usuários devem poder personalizar suas listas com cores, ícones e etiquetas para facilitar a identificação e organização.
8. Backup e Restauração de Dados: O aplicativo deve fornecer recursos de backup e restauração para proteger os dados dos usuários contra perda acidental ou falha do dispositivo.

6. Matriz de Rastreabilidade

Funcionalidade	Req. Usuário	Req. Negócio	Req. Funcionais
<i>1-Interface Intuitiva</i>	<i>Espera uma interface simples e fácil de usar para a criação, edição e compartilhamento de listas.</i>	<i>Proporcionar soluções práticas para as necessidades do dia a dia dos usuários.</i>	<i>1. Registro/Login de Usuário: Permitir que os usuários criem uma conta ou faça login usando suas credenciais existentes. 2. Criação e Edição de Listas: Permitir que os usuários criem novas listas e edite listas existentes de forma intuitiva.</i>
<i>Funcionalidades Versáteis</i>	<i>Deseja poder criar diferentes tipos de listas e organizar listas hierarquicamente.</i>	<i>Facilitar a criação e gerenciamento de listas de forma intuitiva e eficiente.</i>	<i>3. Organização Hierárquica: Suportar a organização hierárquica de listas, permitindo a criação de listas principais e sub listas aninhadas. 4. Personalização de Listas: Permitir que os usuários personalizem suas listas com cores, ícones ou etiquetas.</i>
<i>Facilidade de</i>	<i>Essencial compartilhar</i>	<i>Atingir um amplo</i>	<i>5. Compartilhamento</i>

<i>Compartilhamento</i>	<i>listas com outros usuários, incluindo compartilhamento completo ou de itens individuais.</i>	<i>mercado de usuários que buscam uma solução simples e eficaz para suas necessidades de listas.</i>	<i>de Listas: Permitir que os usuários compartilhem suas listas com outros usuários através de várias plataformas.</i>
-------------------------	---	--	--

7. Descrição dos Casos de Uso

Caso de uso 1

Título	Lista de compras
Id	LC
Descrição	Faz uma lista de compras capaz de colocar os valores dos itens e ter o valor total de todos os elementos da lista.
Atores	Usuários
Pré-condições	Ter criado uma lista com a classificação de lista de compras.
Sequência de tarefa	Criar uma lista no padrão lista de compras, te dará acesso a algumas categorias para organizar sua lista, e ter o valor total gasto, para possuir um controle melhor do financeiro.
Pós-condições	Salvamento da lista após a criação e alteração

Caso de uso 2

Título	Lista de presença
Id	LP
Descrição	Faz uma lista de presença que permite inserir alunos como itens e tirar a falta através de checkboxes
Atores	Usuário
Pré-condições	Criação de uma lista com os campos dos itens necessários
Sequência de tarefa	Criar uma lista no padrão lista de presença, te dará acesso a algumas categorias para organizar sua lista, e ter a possibilidade de fazer a retirada das faltas.
Pós-condições	Salvamento da lista após criação e alteração

Caso de uso 3

Título	Lista de Tarefas
Id	LT
Descrição	Faz uma lista de tarefas que permite inserir tarefas e configurar os campos dos itens conforme desejar
Atores	Usuário
Pré-condições	Criação de uma lista com os campos dos itens necessários
Sequência de tarefa	Criar uma lista no padrão lista de tarefas, com a configuração adequada dos campos dos itens.
Exceções	Salvamento da lista após criação e alteração

8. Protótipos de Tela

Casa

Filtro

Ordenação

Protetor Solar

Descrição

Categoria

Praia

+

Subcategoria

Selecione

+

Data

10 / 02 / 2001

Tempo

04:20:00

Valor

R\$ 12,00

Dicionário

Descrição

Dicionário para o christianzinho

Categoria

Livros

Subcategoria

Crianças

Data

10 / 12 / 2025

Tempo

00:20:30

Valor

R\$ 12,00

Criação da Lista

Título

Descrição

Itens

Selecione os campos a serem adicionados aos itens da sua lista

Descrição

Caixa de seleção

Categoria

Subcategoria

Data

Tempo

Valor

Criar

← Casa

B...

← Filtro

Pesquisar

Limpar

Descrição

Categoria

Praia

Subcategoria

Selecione

Data

10 / 02 / 2001

Tempo

04:20:00

Valor

R\$ 12,00

Dicionário

Descrição

Dicionário para o christianzinho

Categoria

Livros

Subcategoria

Crianças

Data

10 / 12 / 2025

Tempo

00:20:30

Valor

R\$ 12,00

Minhas Listas

Lista de compras

Lista telefonica

Lista de tarefas

Lista de presença

Lista de objetivos

9. Decisões de Arquitetura

Para o desenvolvimento do List Creator App (LCA), foram tomadas diversas decisões arquiteturais visando garantir a eficiência, a escalabilidade e a facilidade de manutenção do aplicativo. As principais decisões incluem:

1. **Arquitetura VMC (View-Model-Controller):** Optou-se por utilizar a arquitetura VMC para separar a interface do usuário (View), a lógica de negócios (Model) e o controle das interações (Controller). Essa abordagem facilita a manutenção e a evolução do aplicativo, além de permitir uma melhor organização do código.
2. **Uso de Android Architecture Components:** Foi implementado o uso dos Android Architecture Components, como ViewModel e LiveData, para garantir uma gestão mais eficiente do ciclo de vida dos componentes da interface do usuário e uma comunicação reativa entre as diferentes partes do aplicativo.
3. **Persistência de Dados com SQLite:** A persistência de dados foi implementada utilizando o SQLite diretamente. O SQLite foi escolhido por sua eficiência e por ser uma solução amplamente utilizada para armazenamento local em aplicativos móveis.
4. **Gerenciamento de Dependências com Dagger:** O Dagger foi escolhido como framework de injeção de dependências para gerenciar a criação e a provisão de componentes do aplicativo. Isso aumenta a modularidade e facilita os testes unitários e de integração.
5. **Arquitetura Modular:** O aplicativo foi dividido em módulos funcionais para garantir a escalabilidade e a reutilização de componentes. Cada módulo trata de uma funcionalidade específica, como gerenciamento de listas, autenticação e compartilhamento, possibilitando que novas funcionalidades sejam adicionadas com mínima interferência no restante do sistema.

10. Decisões de Configuração

Para garantir que o desenvolvimento e a execução do List Creator App (LCA) sejam eficientes e consistentes, foram estabelecidas as seguintes configurações:

1. **Configuração do Ambiente de Desenvolvimento:** O ambiente de desenvolvimento foi configurado utilizando o Android Studio, com a versão mais recente do SDK do Android e as bibliotecas necessárias, como a AndroidX, para assegurar a compatibilidade com as melhores práticas e

recursos atualizados do Android.

2. **Gerenciamento de Versões com Git:** O controle de versão foi realizado através do Git, com repositório hospedado no GitHub. Foram definidos workflows específicos, como a estratégia Git Flow, para organizar o desenvolvimento em branches, facilitando a colaboração e a integração contínua.
3. **Configuração de Build com Gradle:** O Gradle foi utilizado como ferramenta de build, com scripts de configuração para gerenciar as dependências do projeto e automação de tarefas, como minificação de código e geração de APKs.
4. **Configuração de Banco de Dados Local:** O banco de dados local foi configurado utilizando o SQLite. Foram definidos esquemas de tabelas, consultas SQL personalizadas e mecanismos de migração para garantir a integridade e a evolução do esquema de dados.
5. **Autenticação e Segurança:** Para garantir a segurança dos dados dos usuários, foi configurada a autenticação utilizando Firebase Authentication. Além disso, foram implementadas regras de segurança no Firestore para controlar o acesso aos dados armazenados na nuvem.
6. **Integração Contínua:** Foi configurada uma pipeline de integração contínua (CI) utilizando GitHub Actions, para garantir que todos os commits no repositório sejam automaticamente testados e validados, assegurando a integridade do código e a prontidão para o deploy.

11. Ferramentas e Recursos Necessários

1. Ambiente de Desenvolvimento

- **IDE:** Android Studio - Ambiente de desenvolvimento integrado (IDE) para criação de aplicativos Android.

2. Bibliotecas e Frameworks

- **Android SDK:** Conjunto de ferramentas de desenvolvimento para Android.
- **AndroidX Libraries:** Bibliotecas para compatibilidade e componentes adicionais.
- **Material Components for Android:** Para implementação de Material Design (se necessário).
- **Retrofit ou Volley:** Biblioteca para comunicação com APIs RESTful (se necessário).
- **Glide ou Picasso:** Biblioteca para carregamento e manipulação de imagens (se necessário).

3. Design e Layout

- **Design Tool:** Figma ou Adobe XD - Para prototipagem e design de interfaces de usuário.

- **Google Fonts:** Para adicionar fontes personalizadas.
- **Material Icons:** Para ícones padronizados.

4. Controle de Versão

- **Version Control System:** *Git* - Para controle de versão.
- **Repository Hosting:** *GitHub* - Plataforma para hospedagem de repositórios *Git*.

5. Teste e Depuração

- **Testing Frameworks:**
 - **JUnit:** Para testes unitários em Java.
 - **Espresso:** Para testes de UI em Android.
- **Depuração:**
 - **Logcat:** Ferramenta de logs do Android Studio para depuração.
 - **Android Device Monitor:** Para monitoramento de dispositivos e análise de desempenho.

6. Recursos de Backend (se aplicável)

- **Backend Service:** *Firebase* - Para autenticação, banco de dados em tempo real, armazenamento e notificações push (se necessário).

7. Recursos de Aprendizado e Suporte

- **Documentação:**
 - **Android Developer Documentation:** Documentação oficial para desenvolvimento Android.
 - **Java Documentation:** Documentação oficial do Java.
- **Comunidades e Tutoriais:**
 - **Stack Overflow:** Suporte e resolução de problemas.
 - **Google Codelabs:** Tutoriais interativos para desenvolvimento Android.
- **Categorias e Subcategorias:**
 - Definição de modelos de dados (*Categoria*) e a lógica para recuperar e manipular essas informações.
 - Layout XML para a *activity_categoria* e os componentes visuais (*CardView*, *RelativeLayout*, etc.).
- **Navegação:**
 - Configuração de *Intent* para navegação entre *CategoriaTela* e *SubCategoria1*.
- **Imagens e Estilos:**
 - Recursos gráficos (ícones, imagens) mencionados no código (*R.drawable.border*, *R.drawable.prancheta*).
 - Definição de estilos e temas no *styles.xml*.

12.Referências

SITE ANDROID STUDIO (2024) - Site. Disponível em: <https://developer.android.com/studio?hl=pt-br>

SITE FIGMA (2024) - Site. Disponível em:
<https://www.figma.com/files/team/1357464597283449899/recents-and-sharing/recently-viewed?fuid=1357464595273902747>