

Descrição

Público-Alvo/Intended User

Funcionalidades/Features

Protótipo de Interfaces do Usuário

Considerações Chave/Key Considerations

Como seu app vai tratar a persistência de dados?

Descreva qualquer caso de uso específico (“corner case”) da experiência do Usuário (UX).

Descreva quais bibliotecas você utilizará e compartilhe a razão de incluí-las.

Descreva como você implementará o Google Play Services.

Próximos Passos: Tarefas Necessárias

Tarefa 1: Configuração do Projeto/Project Setup

Tarefa 2: Implementar a Interface de Usuário (UI) para cada Activity e Fragment

Tarefa 3: Implementar Persistência de Dados

Tarefa 4: Implementar Activitys e Fragments

Tarefa 5: Configuração para recebimento de imagens

Tarefa 6: Configuração e criação do widget

Usuário do GitHub: <https://github.com/renancacao>

Pocket Memes

Descrição

O Pocket Meme é um novo aplicativo que permite salvar, catalogar e criar novas imagens engraçadas para compartilhar na internet. Você pode encontrar a imagem com a piada certa para o momento certo, enviar as imagens que você recebe pelas redes sociais para o Pocket Memes, para que seja catalogadas e até criar seus próprios memes, pesquisando imagens e adicionando texto.

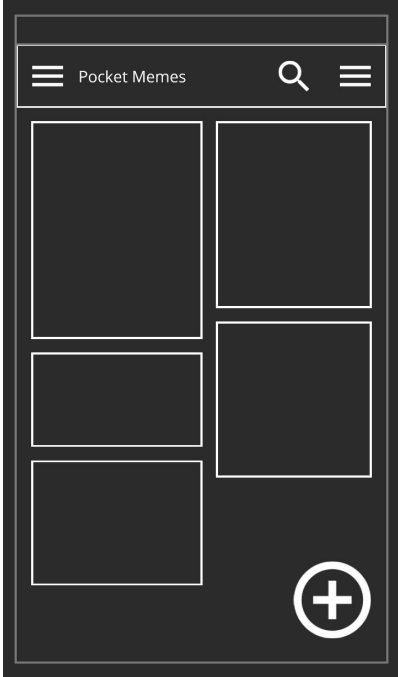

Público-Alvo/Intended User

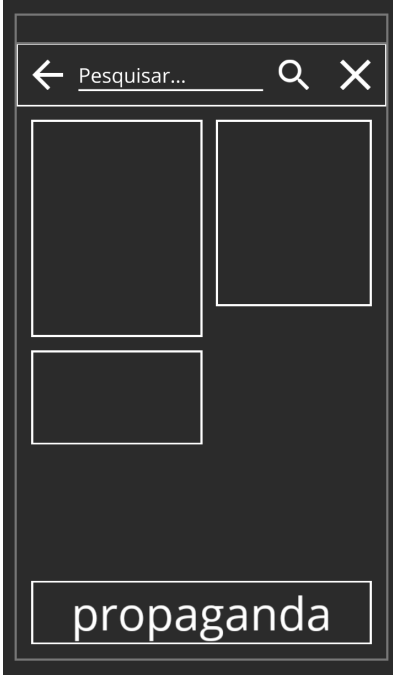

O público alvo deste aplicativo são todos os jovens que hoje utilizam essa nova forma de comunicação pela internet, mais dinâmica, onde uma imagem pode representar tanto, inseridos na cultura das piadas da internet, conhecidas como memes.

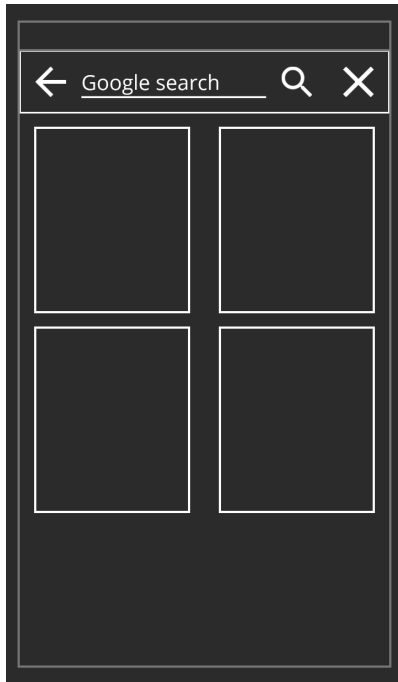
Funcionalidades/Features

- Receber imagens compartilhadas de qualquer outro aplicativo capaz de compartilhá-las.
- Compartilhar os memes cadastrados para qualquer outro aplicativo capaz de receber imagens.
- Catalogar os memes entre grupos.
- Criar novos grupos.
- Escolha de um nome e palavras-chaves para cada imagem.
- Busca de memes por nome ou palavras-chaves.
- Configuração da ordenação de exibição das imagens (Mais novas, mais antigas, mais compartilhadas, por nome).
- Visualização de imagens pertencentes a um grupo específico.
- Pesquisa de imagens na internet.
- Pesquisa de imagens no dispositivo.
- Criação de um novo meme, selecionando uma imagem e adicionando um texto.

Protótipo de Interfaces do Usuário

	<p>A) <u>Tela inicial do aplicativo.</u> Os memes cadastrados aparecem em uma lista com itens de tamanhos não fixos, em duas (retrato) ou três colunas (paisagem).</p> <p>No menu superior temos um botão do lado esquerdo que abre o menu lateral (B), um botão de pesquisa que nos permite pesquisar nossos memes (C) e um botão à direita que nos permite selecionar a ordem de exibição dos memes.</p> <p>O Botão + inferior, quando acionado, nos envia a tela de pesquisa de imagem na web. (G)</p> <p>Quando clicamos em um meme somos enviados a tela de visualização (H)</p>
	<p>B) <u>Menu lateral.</u> Nos permite selecionar o grupo de memes que desejamos visualizar ou criar um novo grupo (E)</p>



	<p><u>C) Campo de pesquisa.</u> Nos permite pesquisar os memes cadastrados por nome ou palavras chave, ao digitarmos o texto e pressionarmos o botão de lupa. O botão X limpa a caixa de texto. Nesta tela será exibida uma propaganda na parte inferior.</p>
	<p><u>E) Novo grupo.</u> Nesta tela podemos selecionar um dos desenhos disponíveis e informar um nome para o novo grupo que será criado.</p>




G) Pesquisa na web. Essa tela utiliza a API Google CustomSearch para pesquisar imagens na web a partir de uma palavra chave. Ao selecionar uma imagem seremos enviados a tela de criação (I).



H) Tela de visualização. Essa tela permite ver os detalhes do meme, como grupos, nome e palavras-chave. Clicando no botão do lápis somos enviados a tela de edição (J). Nesta tela também temos um botão para compartilhamento do meme.

	<p><u>I) Tela de Criação.</u> Nesta tela temos a apresentação da imagem selecionada anteriormente. Podemos informar um texto superior, central e/ou inferior. Clicando nos botões podemos mudar o tamanho da fonte e a cor do texto para tamanhos e cores pré-definidos.</p> <p>Ao clicarmos em Visualizar a imagem exibida é atualizada, mostrando o resultado.</p> <p>Clicando no Ok somos enviados a tela de edição (J)</p>
	<p><u>J) Tela de edição.</u> Na tela de edição podemos definir nome, palavras chaves e os grupos de um meme, seja ele novo ou não. Clicando em OK retornaremos à tela inicial (A).</p>

	<p><u>Widget:</u> O aplicativo conterá um widget que exibirá de forma aleatória, mudando em um determinado tempo, um meme cadastrado na tela do celular do usuário</p>
---	--

Considerações Chave/Key Considerations

Como seu app vai tratar a persistência de dados?

O aplicativo irá gerenciar os dados gerados pelo usuário utilizando-se de um Content Provider. Exemplo desses dados são os grupos, os memes e a relação entre eles. As imagens serão salvas em disco e informações menos importantes, como configurações de ordem de exibição, serão armazenadas com Preferences.

Descreva qualquer caso de uso específico (“corner case”) da experiência do Usuário (UX).

Um dos casos específicos que podemos encontrar no aplicativo, é quando a imagem do meme é apagada do dispositivo, mas ainda é esperada no aplicativo. Neste caso iremos exibir uma imagem padrão que indique este problema ao usuário, de forma que seja fácil a identificação e evite a quebra do fluxo do app.

Descreva quais bibliotecas você utilizará e compartilhe a razão de incluí-las.

Picasso	Para carregar imagens, tanto do dispositivo quanto providas pela internet.
Retrofit	Para gerenciar a comunicação e as respostas da API de busca de imagens.
ButterKnife	Para gerenciar/instanciar as views no código java.
Google Support Library	Material Design.
Google Admob	Para monetização do app.

Google CustomSearch API	Para busca personalizada de imagens
Google Analytics	Para acompanhar e registrar o uso do aplicativo pelos usuários, buscando melhorias.

Descreva como você implementará o Google Play Services.

Utilizaremos o Google Play Services através do Google Admob para exibição de propagandas a fim de monetizar o aplicativo.

A propaganda será exibida como uma faixa na parte inferior de uma tela específica, seguindo as diretrizes do Google Admob.

Utilizaremos também o Google Analytics para realizar o track de uso do aplicativo, utilizando tags pertinentes em passos chave durante o processo de uso, de forma que esses dados nos ajudem a perceber possibilidades de melhorias no uso do aplicativo.

Próximos Passos: Tarefas Necessárias

Tarefa 1: Configuração do Projeto/Project Setup

Configuração das bibliotecas no Grandle.
 Configuração das APIs Externas (API Keys, etc)
 Configuração das permissões no manifesto.
 Internet, Escrita, Leitura, etc.

Tarefa 2: Implementar a Interface de Usuário (UI) para cada Activity e Fragment

Construir UI:
 Tela inicial
 Tela Novo Grupo
 Tela Pesquisa Dispositivo
 Tela Pesquisa Web
 Fragment Exibicao Meme
 Fragment Edit/Visualização de Meme
 Fragment Novo Meme
 Widget

Tarefa 3: Implementar Persistência de Dados

Criar modelos de dados.

Serão criados modelos de dados que facilitem o manuseio dos mesmos. É prevista a criação dos seguintes modelos:

Meme:

Int ID

String Name

String Path

List<String> Tags

List<String> Groupon

Date CreationDate

Date AlterationDate

Group:

Int ID

String Name

String Image

Criar banco de dados.

O banco de dados será criado com auxílio de uma classe extendida de SqliteOpenHelper, que nos auxiliará na administração, criação e alteração do mesmo.

É previsto até o momento a criação de quatro tabelas no banco de dados SQLite.

Uma tabela armazenará os dados dos memes. Outra tabela armazenará os dados dos grupos. Para registrarmos a ligação entre um meme e os vários grupos em que ele pode estar atrelado, utilizaremos uma terceira tabela. De forma semelhante, uma quarta tabela será utilizada para armazenar as palavras chaves de cada meme.

TabMeme:

Int ID *

String Name

String Path

Date CreationDate

Date AlterationDate

TabGroup:

Int ID *
String Name
String Image

TabMemeGroup:
Int ID_meme *
Int ID_group *

TabTag
Int ID_meme *
String tag *

O asterisco (*) representa o campo de chave única da tabela em questão.

Criar Content Provider

A comunicação com o banco de dados em nenhum momento se dará de forma direta. Ao invés disso, teremos um content provider que fornecerá métodos para as principais ações relacionadas ao banco de dados, como inserção, exclusão e busca.

Esse content provider será fechado e apenas nosso aplicativo terá acesso a ele e aos dados do banco de dados.

Tarefa 4: Implementar Activitys e Fragments

Tela inicial

- Lista de memes
- Fab buttons
- Toolbar
- Pesquisa
- Menu Ordenação
- Menu Lateral
- Lista de grupos
- Exclusão de grupos
- Exibição de propaganda quando necessário.

Tela Novo Grupo

- Lista de imagens
- Cadastro

Tela Pesquisa Dispositivo

- Receber lista de imagens do dispositivo
- Lista de imagens

Tela Pesquisa Web

- Chamada de API com LoaderAsyncTask
- Lista de imagens

Fragment Exibicao Meme

- Utilização do Picasso para carregar/editar imagem

Fragment Edit/Visualização de Meme

- Mudança entre exibição e edição.
- Cadastro/busca de informações

Fragment Novo Meme

- Envio de informações para edição de imagem

Widget

- Atualização automática
- Exibição aleatória

Tarefa 5: Configuração para recebimento de imagens

Configuração de Intent Filters, etc.

Tarefa 6: Configuração e criação do widget

Criação e configuração de um widget que exibirá um meme aleatório a cada atualização.