

[Descrição](#)

[Público-Alvo/Intended User](#)

[Funcionalidades/Features](#)

[Protótipo de Interfaces do Usuário](#)

[Navigation Drawer](#)

[Main Activity](#)

[Game Details Activity](#)

[Details tab](#)

[Favorites Activity](#)

[Wishlist Activity](#)

[Widget](#)

[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 as models e classes para lidar com a API](#)

[Tarefa 4: Persistência de dados](#)

[Tarefa 5: Implementar widget](#)

[Tarefa 6: Google Play Services](#)

[Implementar Admob.](#)

[Tarefa 7: Testes](#)

[Implementar testes automatizados.](#)

[Tarefa 8: Preparar para lançamento](#)

Usuário do GitHub: Gabriel-Agrelli

Game Center

Descrição

Com o app Game center é possível buscar e visualizar uma variedade de games de todas as plataformas, de forma simples e rápida. Visualize os jogos mais populares, lançamentos e os mais aguardados pelos gamers. Marque os seus jogos favoritos e compartilhe com seus amigos, adicione-os também em sua lista de desejos.

Todos as informações relacionadas aos jogos são fornecidas pelo [IGDB API](#).

Público-Alvo/Intended User

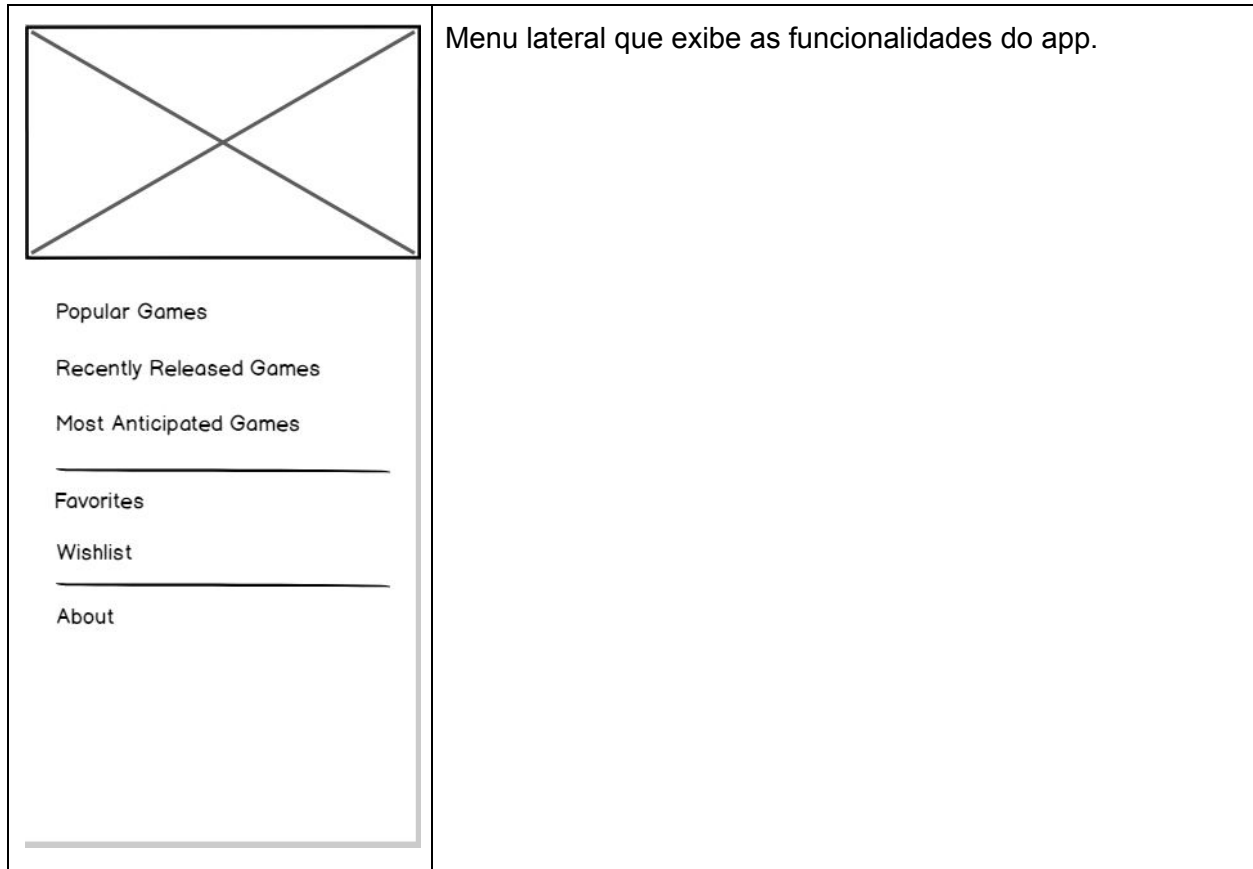
Este app é destinado para os fãs de jogos eletrônicos que queiram acompanhar os lançamentos e tendências da indústria de games.

Funcionalidades/Features

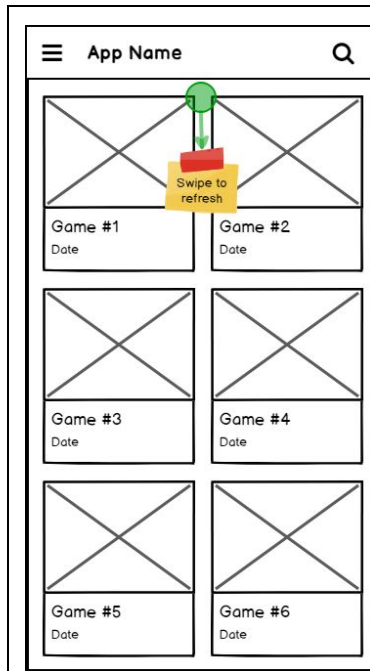
- Busca por games
- Marcar games favoritos
- Compartilhar as informações de games
- Lista de desejos
- Widget

Protótipo de Interfaces do Usuário

Navigation Drawer



Main Activity

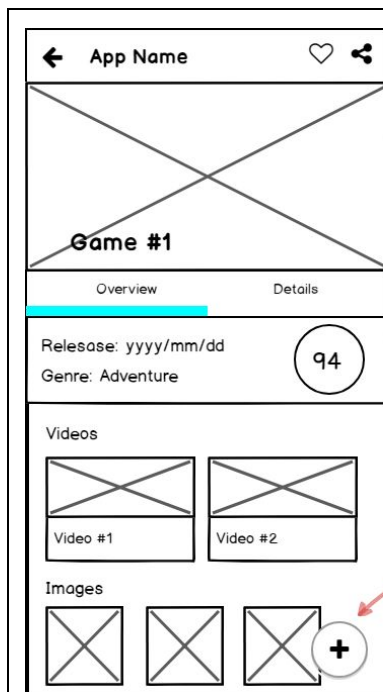


Tela principal da aplicação que lista os jogos populares no momento.

Contém um campo para busca de jogos.

Implementa swipe to refresh para atualizar a lista de jogos.

Game Details Activity



Tela de detalhes, onde é listado todas as informações do jogo selecionado.

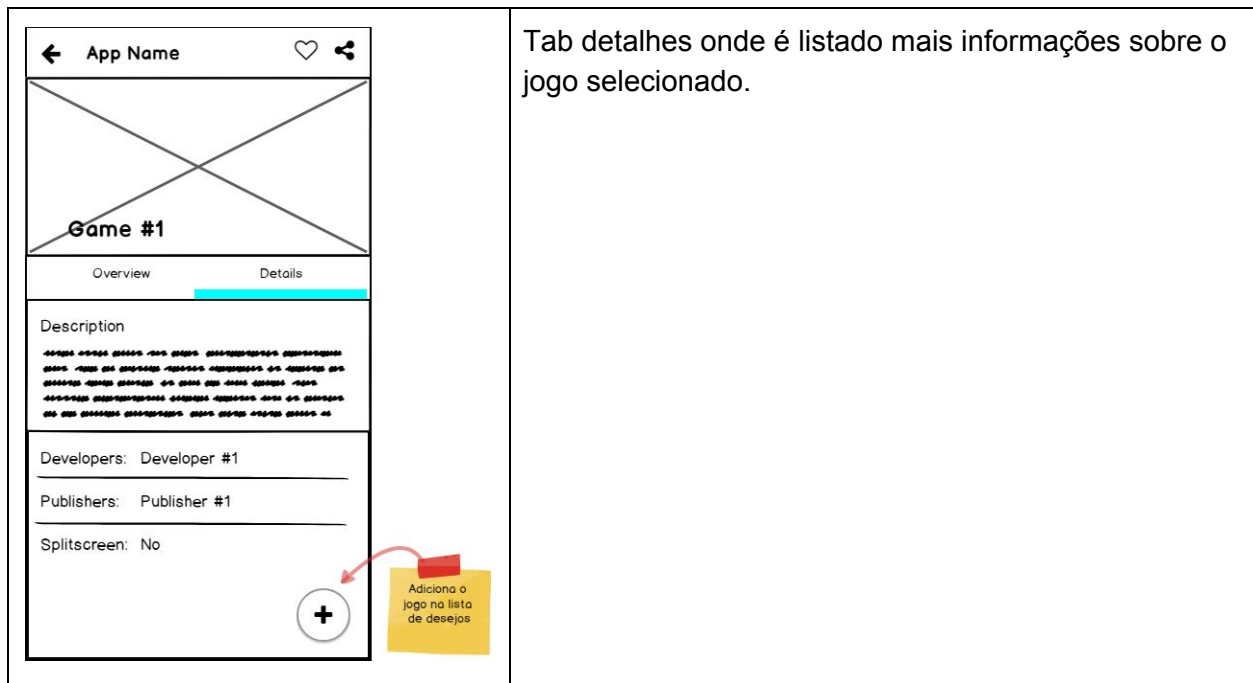
É possível adicionar o jogo aos favoritos, assim como também adicioná-lo à lista de desejos através do floating action button.

Há um tab bar com 2 seções:

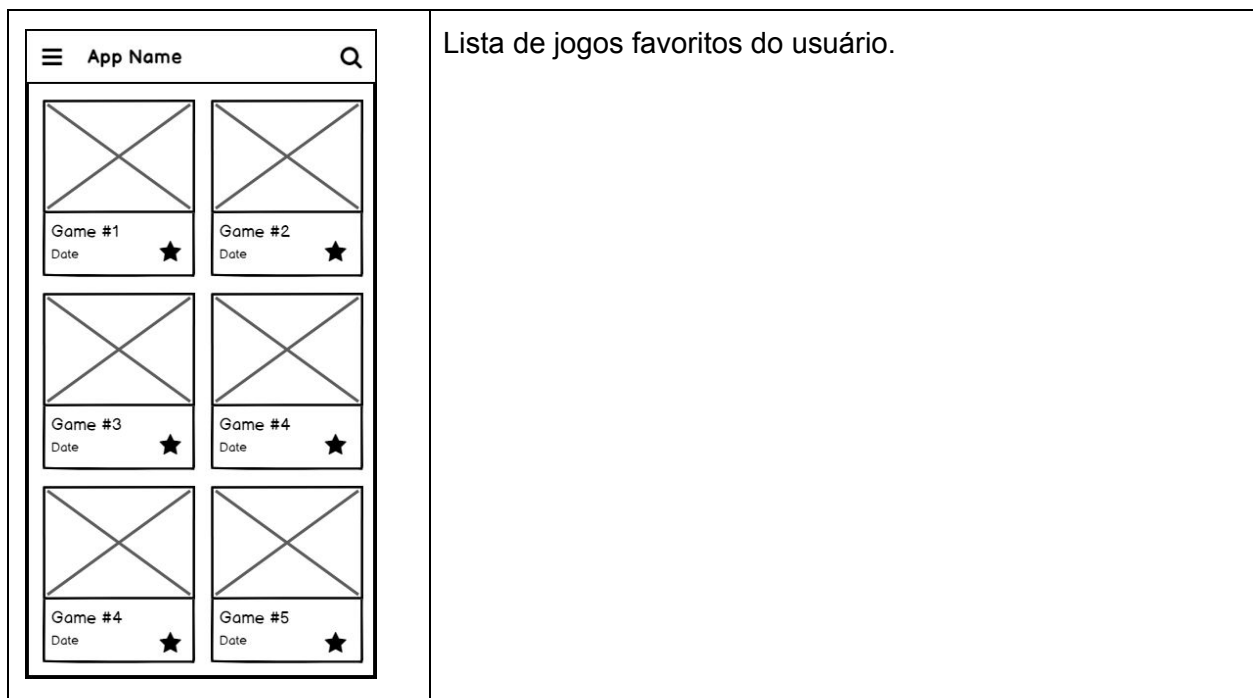
Overview: é mostrado algumas informações sobre o jogo, como a sua nota, data de lançamento, imagens e vídeos.

Details: é mostrado a descrição do jogo e informações sobre desenvolvedora, publicadora e informação sobre se possui o modo de jogo 'tela dividida'.


Details tab



Favorites Activity



Wishlist Activity

<div>  App Name </div> <hr/> <div> <input type="radio"/> Game #1 </div> <hr/> <div> <input type="radio"/> Game #2 </div> <hr/> <div> <input type="radio"/> Game #3 </div> <hr/> <div> <input type="radio"/> Game #4 </div> <hr/>	<p>Lista de desejos.</p> <p>A lista de desejos é uma espécie de lista de compras, no qual o usuário pode selecionar aquele jogo no qual ele já adquiriu.</p>
---	--

Widget

<div>Lista de desejos</div> <hr/> <div> <input type="radio"/> Game 1 </div> <div> <input type="radio"/> Game 2 </div> <div> <input type="radio"/> Game 3 </div> <div> <input type="radio"/> Game 4 </div>	<p>Widget lista de desejos.</p> <p>Lista de jogos que o usuário adicionou em sua lista de desejos.</p> <p>Usuário poderá selecionar o radio button para marcar um jogo no qual ele já adquiriu.</p>
---	---

Considerações Chave/Key Considerations

Como seu app vai tratar a persistência de dados?

O app irá armazenar os games favoritos e a lista de desejos no SQLite usando um Content Provider.

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

- **Sem conexão com a internet:** Uma mensagem será exibida ao usuário de que é necessário se conectar a internet.
- **Mudança de orientação:** Deverá lidar de forma correta com operações que estejam sendo executadas em background e armazenar o estado atual da activity com o método [onSaveInstanceState\(\)](#).

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

- **Butter Knife:** para facilitar o binding dos componente visuais
- **Picasso:** para lidar com o carregamento de imagens
- **Retrofit:** para a comunicação com a API
- **Gson:** para a serialização e desserialização de JSON
- **AdMob:** para monetização

Descreva como você implementará o Google Play Services.

O app utilizará o Ad Mob para monetização.

Próximos Passos: Tarefas Necessárias

Tarefa 1: Configuração do Projeto/Project Setup

- Criar novo projeto no Android Studio
- Adicionar as dependências das bibliotecas que serão utilizadas
- Criar a estrutura básica de pastas

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

- Construir a UI de todas as activities e fragments
- Construir o Navigation Drawer
- Baixar ícones que serão utilizados no app
- Escolher esquema de cores

Tarefa 3: Implementar as models e classes para lidar com a API

Classes necessárias:

- Models
- Loaders
- Adapters
- Interfaces

Tarefa 4: Persistência de dados

Implementar Content Provider e Shared Preferences helpers para lidar com os dados que serão armazenados localmente.

Tarefa 5: Implementar widget

Construir a UI do widget e implementar sua lógica.

Tarefa 6: Google Play Services

Implementar Admob.

Tarefa 7: Testes

Implementar testes automatizados.

Tarefa 8: Preparar para lançamento

- Criar ícone do app
- Revisar código
- Remover logs (caso exista)