

[Descrição](#)

[Público-Alvo/Intended User](#)

[Funcionalidades/Features](#)

[Protótipo de Interfaces do Usuário](#)

[Tela 1](#)

[Tela 2](#)

[Tela 3](#)

[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](#)

[Tarefa 2: Implementar um wrapper para a API](#)

[Tarefa 3: Implementar a Interface de Usuário \(UI\) para cada Activity e Fragment](#)

[Tarefa 4: Implementar o ContentProvider](#)

[Tarefa 5: Implementar as notificações](#)

[Tarefa 6: Implementar o widget](#)

[Tarefa 7: Implementar o Firebase](#)

Usuário do GitHub: lipeduoli

Run Speed, Run

Descrição

Speedrun é uma categoria nos jogos onde a intenção é terminar o jogo da forma mais rápida possível, almejando entrar no ranking dos melhores de cada jogo.

Neste app você poderá ver quem são os mais rápidos nas categorias de diversos jogos. Acompanhe os melhores jogadores e seja notificado caso alguém pegue a dianteira na sua categoria favorita.

Público-Alvo/Intended User

O público alvo do aplicativo são gamers e entusiastas de SpeedRuns, uma categoria nos jogos onde é preciso ser o mais rápido possível para completar o jogo.

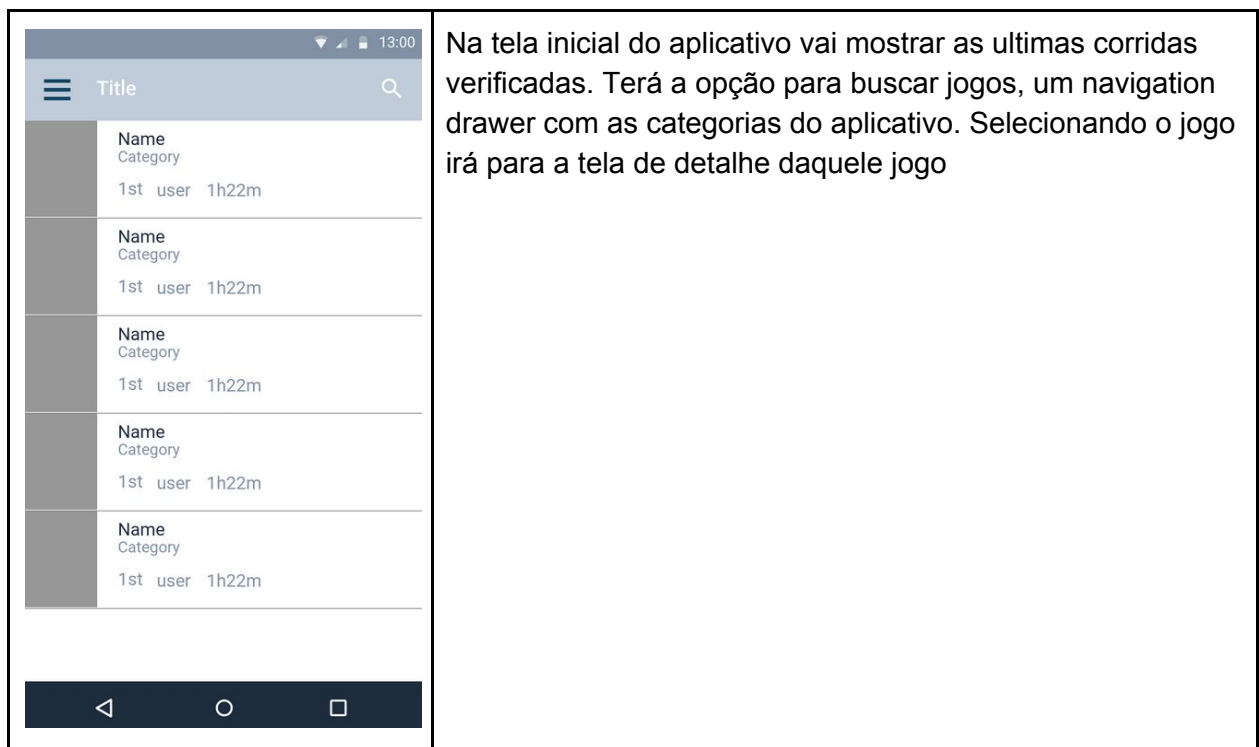
Funcionalidades/Features

- O aplicativo usará a API do site www.speedrun.com
- Visualize as corridas mais recentes
- Acompanhe a classificação do seu jogo favorito
- Salve um jogo/categoria favorita e seja notificado caso alguém pegue o primeiro lugar.

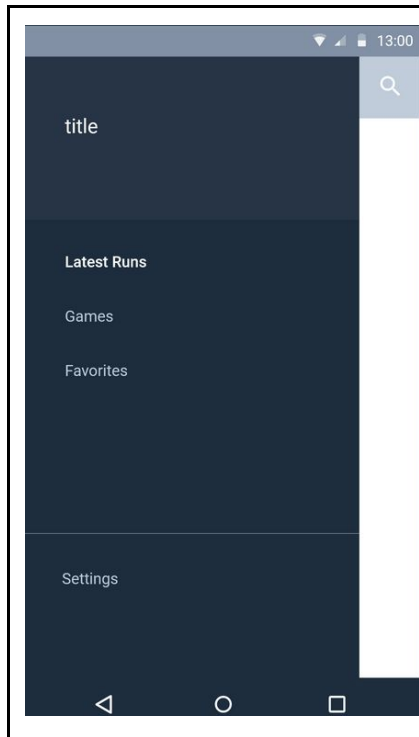
Protótipo de Interfaces do Usuário

Elas podem ser feitas a mão (tire uma foto dos seus desenhos e os insira neste fluxo), ou usando um programa como o Photoshop ou Balsamiq.

Tela 1



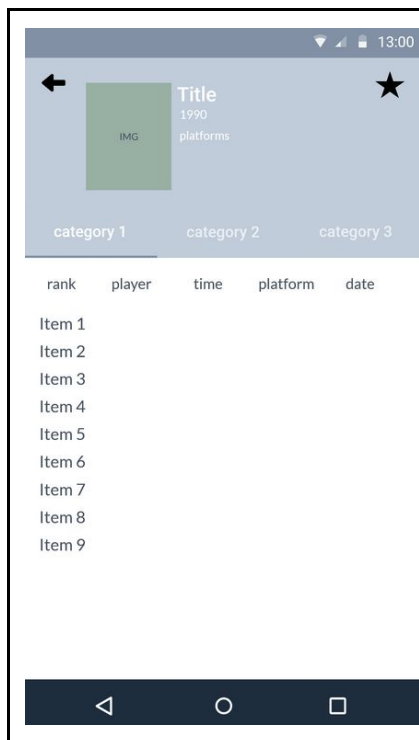
Tela 2



O navigation drawer irá conter as áreas do aplicativo que serão:

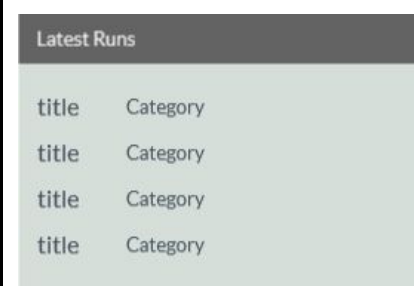
- Latest Runs com as ultimas corridas verificadas
- Games podendo pesquisar qualquer jogo
- favoritos que será mostrado os jogos favoritos do usuario.

Tela 3



A tela de detalhe do jogo irá conter a imagem do jogo e alguns detalhes, uma tab que poderar rolar mostrando as categorias do jogo e ao selecionar uma categoria é exibido a classificação daquela categoria. Clicando em algum item da classificação será exibido o vídeo da corrida, caso tenha. Terá também a opção para salvar nos favoritos aquela categoria do jogo que estiver selecionado.

Widget

	Widget onde será mostrado as corridas mais recentes
---	---

Considerações Chave/Key Considerations

Como seu app vai tratar a persistência de dados?

O aplicativo terá um ContentProvider que será usado para salvar os favoritos do usuário. Será salvo o Id e nome do Jogo e da Categoria e o Id da corrida que estiver em primeiro lugar.

Esses dados serão usados na tela de detalhe do jogo alterando o ícone de favoritos caso a categoria esteja salva e no JobDispatcher, onde será verificado cada jogo salvo se tem um record novo para disparar uma notificação, caso seja verdadeiro.

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

O usuário poderá marcar uma ou mais categorias de algum jogo como favorito pela activity de detalhe do jogo. Para remover este favorito o usuário deve acessar o navigation drawer pela tela inicial e selecionar a categoria Favoritos.

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

- Retrofit 2, para tratar da comunicação com a API
- Glide, para tratar o carregamento e cache das imagens.
- Palette, será usado para configurar a cor da interface da tela de detalhes com as cores da capa do jogo.
- Design, biblioteca de suporte do android para a montagem do layout
- RecyclerView, biblioteca de suporte do android para a criação das listas
- ConstraintLayout, usado na criação dos layouts

Descreva como você implementará o Google Play Services.

Serão usados os seguinte serviços do Firebase:

- Google Analytics para acompanhar os dados de uso do aplicativo.
- AdMob para exibir anuncios no formato de banner no aplicativo.
- Crashlytics para acompanhar possíveis erros que ocorram com os usuarios.

Próximos Passos: Tarefas Necessárias

Esta é a parte onde você falará sobre as principais funcionalidades do seu app (mencionadas acima) e as dividirá em tarefas técnicas tangíveis que você pode completar de forma incremental até finalizar o app.

Tarefa 1: Configuração do Projeto

- Criar o projeto
- Configurar as bibliotecas

Tarefa 2: Implementar um wrapper para a API

- Criar uma biblioteca Java
- Configurar as bibliotecas
- Criar services para cada parte da API que será usada

Tarefa 3: Implementar a Interface de Usuário (UI) para cada Activity e Fragment

- Criar a UI de cada activity/fragment
- Implementar cada activity/fragment

Tarefa 4: Implementar o ContentProvider

- Criar o ContentProvider
- Implementar a funcionalidade para salvar os jogos/categoria nos favoritos

Tarefa 5: Implementar as notificações

- Criar um JobDispatcher que verifique as categorias favoritas e notifique caso tenha mudança na classificação

Tarefa 6: Implementar o widget

- Criar o layout do widget
- Implementar a funcionalidade do widget listar as corridas mais recentes

Tarefa 7: Implementar o Firebase

- Implementar o Google Analytics
- Implementar o Crashlytics
- Implementar o AdMob