

Como utilizar este Template

1. Faça uma cópia [File → Make a copy...]
2. Renomeie este arquivo para: **“Capstone_Stage1”**
3. Substitua todos os textos **em verde**

Instruções para Envio

1. Assim que completar todas as partes, faça o download deste documento como um PDF [File → Download as PDF]
2. Crie um repositório novo no Github para o capstone. Dê o nome **“Capstone Project”**
3. Adicione este documento no seu repositório. Tenha certeza que está nomeado como **“Capstone_Stage1.pdf”**

[Descrição](#)

[Público-Alvo/Intended User](#)

[Funcionalidades/Features](#)

[Protótipo de Interfaces do Usuário](#)

[Tela 1](#)

[Tela 2](#)

[Tela 3](#)

[Tela 4](#)

[Tela 5](#)

[Tela 6](#)

[Tela 7](#)

[Tela 8](#)

[Tela 9](#)

[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: Criar serviço para recuperar as informações da API da Blizzard](#)

[Tarefa 4: Cacheamento das informações](#)

[Tarefa 5: Salvamento remoto das informação](#)

[Tarefa 6: Adicionar tratamentos de erro](#)

[Tarefa 7: Implementar Widget](#)

[Tarefa 8: Revisar e aplicar alterações para acessibilidade](#)

Usuário do GitHub: [duccipopi](#)

Guild Herald for WoW

Descrição

Veja, compare e gerencie sua guilda de World of Warcraft!

Acompanhe o progresso e evolução dos seus companheiros de guilda, ou adicione um novo nível de competição amigável através do progresso das masmorras, raides e equipamento.

Público-Alvo/Intended User

Jogadores de World of Warcraft que desejam adicionar um novo nível de competição em suas guildas.

Funcionalidades/Features

- Veja as informações dos seus personagens de World of Warcraft
- Veja as informações da sua guilda e companheiros
- Acompanhe o progresso da guilda
- Defina ranques e critérios para comparação entre personagens

Protótipo de Interfaces do Usuário

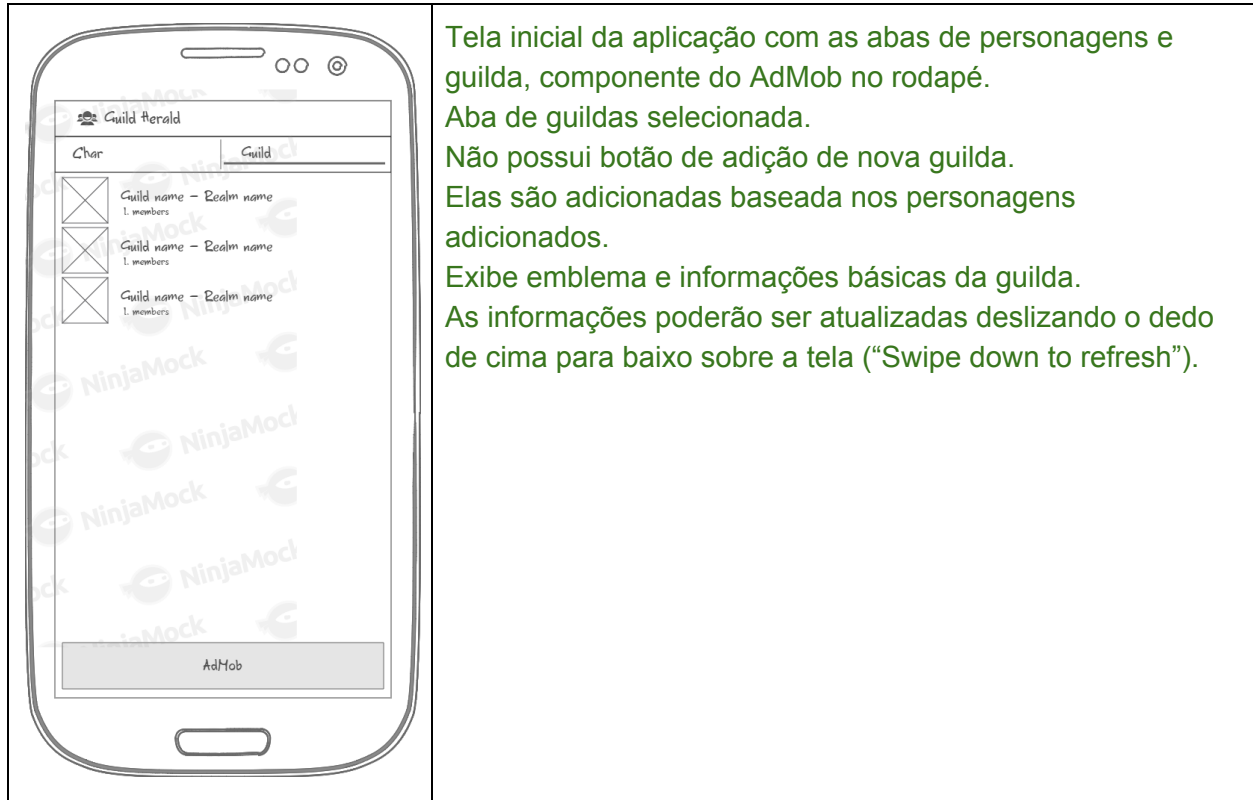
Elas podem ser feitas a mão (tire uma foto dos seus desenhos e os insira neste fluxo), ou usando um programa como o Photoshop ou Balsamiq.

Tela 1



Tela inicial da aplicação com as abas de personagens e guilda e componente do AdMob no rodapé.
Aba de personagens selecionada exibindo botão de adição de novo personagem, imagem e informações básicas dos personagens já adicionados.
As informações poderão ser atualizadas deslizando o dedo de cima para baixo sobre a tela ("Swipe down to refresh").

Tela 2



Tela 3



Tela de adição de personagem com componente do AdMob no rodapé.

Busca exige nome e reino do personagem.

Resultado exibe imagem e informações básicas do personagem.

A API só permite busca exata, fazendo com que sempre seja retornado apenas um resultado.

Tela 4



The mockup shows a mobile application interface for character details. At the top, there's a header with a placeholder for a character's portrait. Below this, a list of attributes is displayed: Race, Class Level, Item Level, Achievements, Honorable kills, Guild name, and Guild Realm. A section titled 'Character name - Realm' follows. Below that is a calendar view showing the current day, week, month, and agenda. At the bottom, there's an equipment list with three slots, each showing a placeholder for an item and its name.

Tela de detalhes de personagem.
Exibe imagem, informações, atributos e equipamento do
personagem selecionado.

Tela 5



Tela 6



Tela de detalhes da guilda com a lista de membros da guilda selecionada, ordenada por ranque dentro da guilda. Exibe emblema e informações da guilda, além de diferentes ordenações para os membros.

Tela 7



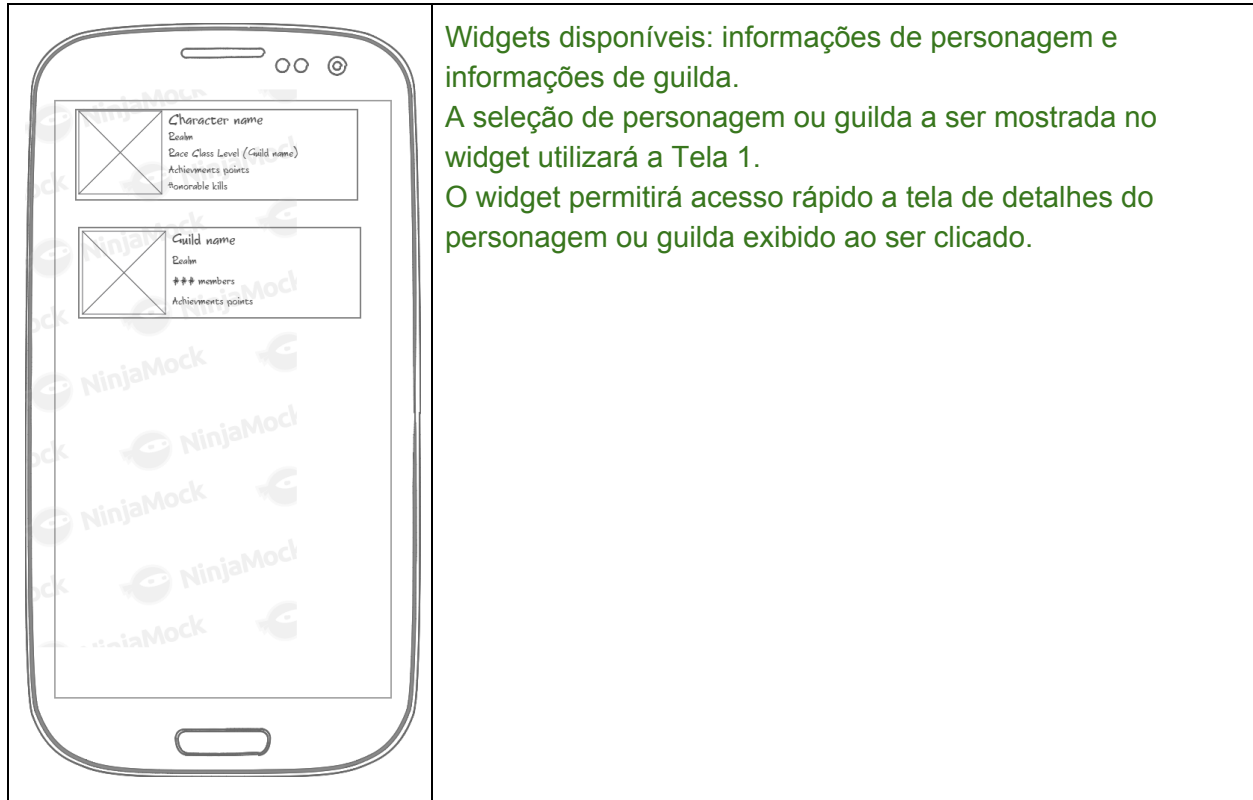
Tela 8



The mockup shows a mobile application interface for setting custom ranks. At the top, there's a title bar with a back arrow, a user icon, the text 'Set custom rank', and a checkmark. Below this is a form with three input fields: 'Rank name', 'Stat', and 'Min value'. To the right of these fields is a large plus sign. Below the form is a list of five entries. Each entry consists of a 'Rank name' field, a 'Statistic' field, a '> Min value' label, and a delete icon (an 'X'). At the bottom of the screen is an AdMob banner.

Tela de configuração de ranque customizado.
Permite adicionar diversos ranques baseado em estatísticas de personagem.

Tela 9



Considerações Chave/Key Considerations

Como seu app vai tratar a persistência de dados?

Content provider para persistência local de dados e Firebase para persistência na nuvem.
 A **persistência local** evitará acesso excessivo à API e uso de dados, já que a consulta a ela possui limitação de requisições (por hora e por minuto). A quota de uso e tempo de renovação da quota são enviados no cabeçalho de toda resposta da requisição, assim é possível determinar quando utilizar as informações locais em detrimento de fazer uma nova requisição. A arquitetura do projeto deve tornar transparente as camadas superiores (UI) a origem do dado através de uma interface de acesso de dados, que por sua vez será responsável por decidir entre utilizar a base de dados local ou requisitar atualização através de requisição via API. A persistência local também deve armazenar os personagens e guildas adicionados pelo usuário, além do ranque customizado para cada guilda.
 O **Firebase** será utilizado para persistência dos personagens e guildas adicionados pelo usuário, além do ranque customizado para cada guilda, possibilitando a transferências e sincronia dessas informações entre dispositivos.

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

Deve ser possível sempre ir a tela de detalhes de personagem quando se clicar no retrato do mesmo ao longo de todas as telas

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

Picasso para tratar o carregamento e cache das imagens.

Retrofit para acesso e consumo da rest API da Blizzard.

Descreva como você implementará o Google Play Services.

Firebase: Persistência da preferência do usuário e ranque customizado dos membros da guilda.

AdMob: Uso de propagandas no rodapé das telas e propaganda intersticiais quando voltando das tela de detalhes (guilda e personagem).

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/Project Setup

- Configurar bibliotecas
- Criar as contas de acesso das API
- Externalizar as chaves das API

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

- Construir a UI da Activity principal, contendo as abas de lista de personagem e clãs
- Construir a UI da Activity de detalhes de personagem
- Construir a UI da Activity de detalhes do clã
- Construir a UI do Fragment de lista de membros do clã
- Construir a UI do Fragment de lista de ranques, mostrando os personagens em cada um
- Construir a UI da Activity de configuração de ranques

Tarefa 3: Criar serviço para recuperar as informações da API da Blizzard

Implementar o serviço de obtenção de informações dos personagens e clã.

- Baixar as informações via Retrofit
- Criar interfaces e objetos de dados para uso das informações

Tarefa 4: Cacheamento das informações

Salvar e cachear localmente as informações para reduzir consultas a API e proporcionar usabilidade em modo desconectado/offline.

- Criar Content Provider
- Adicionar camada de decisão de escolha de origem da informação

- Alterar UI para informar falta de conexão de dados
- Alterar UI para possibilitar atualização forçada dos dados

Tarefa 5: Salvamento remoto das informação

Salvamento remoto das configurações de ranque e dados de usuário.

- Implementar banco de dados do Firebase
- Implementar interface para uso do Firebase
- Sincronizar informações locais e remotas

Tarefa 6: Adicionar tratamentos de erro

Verificar e implementar tratamentos de erro adequados.

- Implementar validações para erros de conexão
- Implementar validação de consistência de dados
- Adicionar informações de UI para erros relevantes ao usuário

Tarefa 7: Implementar Widget

Implementar widget para exibição de informação de personagem ou guilda.

- Implementar UI do Widget
- Implementar UI de configuração do Widget

Tarefa 8: Revisar e aplicar alterações para acessibilidade

Revisar e alterar as implementações e UI para possibilitar acessibilidade

- Remover qualquer *string* fixa do código e UI, permitindo traduções
- Adicionar descritivo as imagens
- Revisar e alterar o fluxo de navegação entre elementos da UI, tornando fluído o uso do D-Pad

Adicione quantas tarefas que achar necessárias para concluir seu app.

Instruções para Envio

4. Assim que completar todas as partes, faça o download deste documento como um PDF [File → Download as PDF]
5. Crie um repositório novo no Github para o capstone. Dê o nome “**Capstone Project**”
6. Adicione este documento no seu repositório. Tenha certeza que está nomeado como “**Capstone_Stage1.pdf**”