

## Sumário

[Descrição](#)

[Usuários](#)

[Funcionalidades](#)

[Protótipo da Interface com o usuário](#)

[Tela 1 – Tela principal](#)

[Tela 2 – Lista localizações favoritas](#)

[Tela 3 – Tela de login](#)

[Tela 4 – Notificações](#)

[Considerações chaves](#)

[Descreva qualquer caso excepcional, se houver, a respeito da experiência do usuário \(UX – User eXperience\).](#)

[Descreva as bibliotecas e/ou componentes externos que você está pensando em utilizar e forneça as razões para incluí-las no seu projeto.](#)

[Planejamento das tarefas da Sprint 01](#)

**Repositório GitHub:** <https://github.com/carlosolimpio/WhattoWeather>

**Protótipo no Marvel:** <https://marvelapp.com/prototype/7221c71/screen/74897702>

**Equipe:** Carlos Olimpio

# WhattoWeather app

## Descrição

What to Weather é um aplicativo de tempo onde você pode verificar a previsão do tempo como muitos outros apps. A ideia principal é mostrar uma peça de roupa que você pode usar como sugestão para o clima que está sendo previsto para uma determinada localização.

## Usuários

O principal público do aplicativo será o público feminino jovem que se interessa por moda, que gosta de estar atualizado sobre o clima e como se vestir nesse clima.

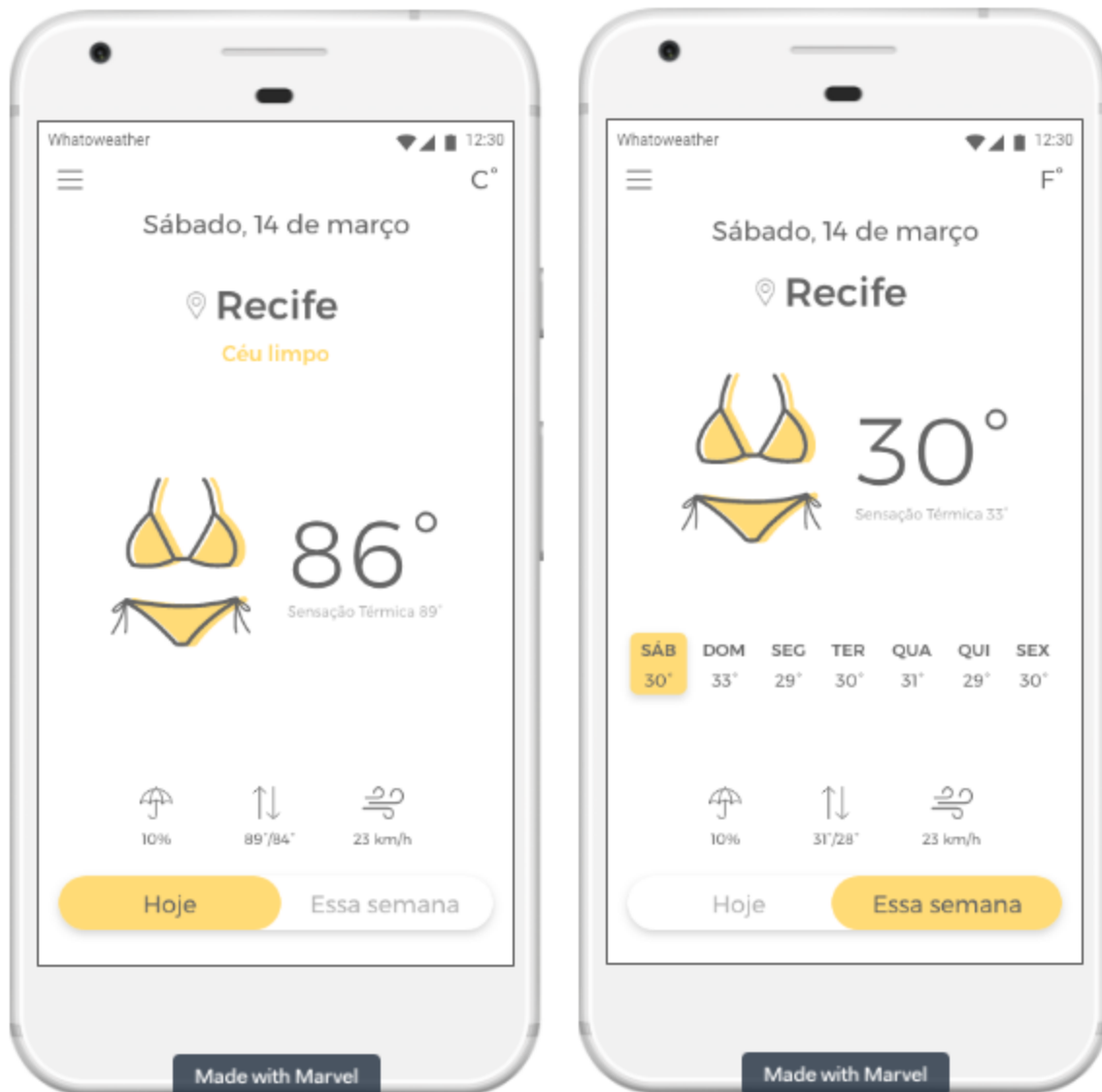
## Funcionalidades

As principais funcionalidades da aplicação são:

- Mostrar informações sobre o clima atual (do dia) e o clima da semana seguinte (próximos sete dias). As informações do clima serão:
  - Temperatura
  - Descrição do clima
  - Ícone com a roupa recomendada para o clima
  - Sensação térmica
  - Porcentagem de precipitação
  - Temperaturas máxima e mínima
  - Velocidade do vento
- Possibilidade para troca de graus Celsius para Fahrenheit
- Busca do clima para uma localização específica
- Possibilidade para adição/remoção de localidades como favorito bem como mostrar sua temperatura de maneira rápida
- Social login
- Notificação, caso o clima mude, mostrando informações relevantes como o novo ícone, temperatura e descrição do clima
- Compartilhamento do clima atual em redes sociais

## Protótipo da Interface com o usuário

### Tela 1 - Tela principal



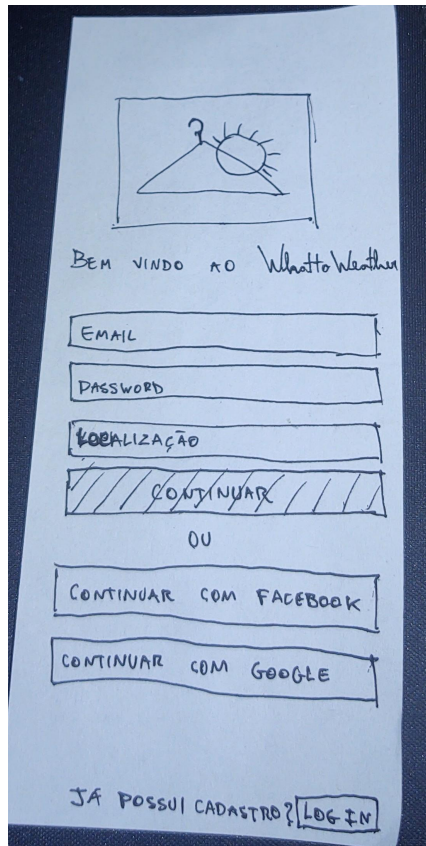
Na tela principal, o usuário tem acesso à principal feature do app: temperatura, descrição do clima, ícone com peça de roupa e outras descrições do clima. Além disso, o usuário poderá ver como está o clima para os próximos 7 dias.

## Tela 2 - Lista localizações favoritas



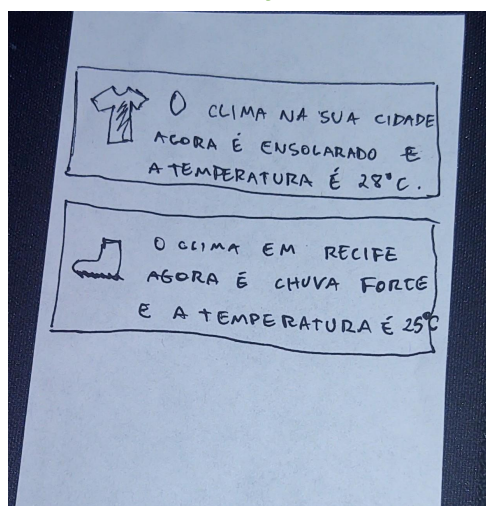
Nesta tela o usuário pode ver sua lista de localizações salvas, bem como adicionar e excluir. Além disso, pode ter uma rápida noção sobre o clima dessas localizações salvas.

### Tela 3 - Tela de login



Nesta tela de login, o usuário pode logar usando o login social com google ou facebook ou pode optar por logar usando credenciais normais.

### Tela 4 - Notificações



As notificações serão mostradas a cada mudança de clima, se o usuário permitir essa configuração, mostrando o ícone, temperatura e descrição do clima. Se clicadas, levarão à página principal do app.

## Considerações chaves

### Como seu app vai lidar com a persistência de dados?

A aplicação contará com o backend na nuvem utilizando o Firebase para os dados de login e localizações favoritas do usuário. Para o cache dos dados será utilizado o Room onde serão armazenados esses dados.

### Descreva qualquer caso excepcional, se houver, a respeito da experiência do usuário (UX - *User eXperience*).

- Caso o usuário esteja na tela inicial, pressionar o botão de voltar, o app será minimizado
- Caso o usuário esteja adicionando uma nova localização e pressionar o botão de voltar, o app voltará para a lista de favoritos
- O app perguntará ao usuário para utilizar sua localização atual via API do smartphone, caso negue o app dará opção do usuário adicionar manualmente sua localização
- Caso o app não consiga encontrar o clima para uma localização específica, será mostrada uma caixa de diálogo informando o erro para o usuário
- Caso o app seja aberto e não esteja conectado à internet, será mostrado uma caixa de diálogo pedindo para o usuário ligar seu Wi-Fi ou seus dados móveis.
  - Caso o usuário persista em usar o app sem estar conectado à internet, o app tentará buscar na memória do dispositivo a última previsão salva. Se esta for mais velha do que 1 dia, o app mostrará uma tela de erro de sincronização. Se esta for coletada em menos de 24 horas, o app mostrará aqueles dados.

### Descreva as bibliotecas e/ou componentes externos que você está pensando em utilizar e forneça as razões para incluí-las no seu projeto.

- Retrofit para consumo de APIs de clima
- Room para o cache de dados locais
- Google Geolocation API para acesso a geolocalização
- Google Sign-in para o social login
- Firebase para backend

## Planejamento das tarefas da Sprint 01

- Implementação da arquitetura do app utilizando MVVM
- Construção inicial das telas 1 e 2 com Android jetpack compose
- Implementação para consumo das APIs de clima com Retrofit
- Implementação da camada de dados (model) para persistir os dados com Room