

# PROJETO MOBILE

Professor: João Ferreira

Data de entrega: 25/11/2024

App de Monitoramento Climático com Ionic



# UNINASSAU

Uninassau-Graças

## EQUIPE

Carlos Alberto Ramalho Bezerra Neto  
Matrícula — 01585045

José Gabriel de Oliveira Lino  
Matrícula — 01609620

Márcio Cavalcanti Sobel  
Matrícula — 01578025

Gustavo portela Pachêco  
Matrícula — 01604533

# GOOGLE CALENDAR(Agenda)

14

Agenda

Hoje

<

>

Novembro de 2024

🔍

?

⚙️

Mês ▾

📅

🕒

⋮

+

Criar ▾

Novembro de 2024

<

>

D	S	T	Q	Q	S	S
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
1	2	3	4	5	6	7

Pesquisar pessoas

Minhas agendas

^

☒

Carlos Ramalho

☐

Aniversários

☐

Tarefas

Outras agendas

+ ^

☐

Eletrotécnica 2021.1 E3

☐

Feriados no Brasil

☐

Holidays in Brazil

☐

Turma T4 - Cálculo Numéri...

Termos - Privacidade

DOM. 27	SEG. 28	TER. 29	QUA. 30	QUI. 31	SEX. 1 nov.	SÁB. 2
	Conversa Sobre Proj. Mob				Primeiras entregas Mobile	
3	4	5	6	7	8	9
		11:30am Reunião pós aula				
10	11	12	13	14	15	16
		Reunião Mobile				
17	18	19	20	21	22	23
	Reunião Mobile				Feedback Mobile	
24	25	26	27	28	29	30
	Mobile Weekly				ÚLTIMA REUNIÃO PRÉ EN*	

Mostrar área de

# ESTRUTURA ANALITICA DO PROJETO

## EAP

Estrutura Analítica do Projeto (EAP) - App de Monitoramento Climático com Ionic

### Planejamento

- 1.1. Definição dos Requisitos do Projeto
- 1.2. Identificação das APIs REST Climáticas
- 1.3. Planejamento do Cronograma e Recursos
- 1.4. Análise de Riscos

### Design e Prototipagem

- 2.1. Desenvolvimento do Design UI/UX
- 2.2. Validação do Protótipo com Stakeholders

### Configuração do Ambiente de Desenvolvimento

- 3.1. Instalação e Configuração do Ionic Framework
- 3.2. Configuração do Ambiente de Desenvolvimento Android
- 3.3. Configuração e Testes de Integração com APIs REST

### Desenvolvimento

- 4.1. Desenvolvimento do Backend e Integração com APIs
  - 4.1.1. Consumo da API de Clima Pública
  - 4.1.2. Tratamento de Dados (Parse e Formatação)
- 4.2. Desenvolvimento do Frontend
  - 4.2.1. Criação da Tela Inicial e Detalhes
  - 4.2.2. Implementação de Componentes de Interface
  - 4.2.3. Exibição de Dados Climáticos (Atuais e Previsões)
- 4.3. Implementação de Funcionalidades Adicionais
  - 4.3.1. Busca de Cidades

## Testes

5.1. Testes Unitários

5.2. Testes de Integração

5.3. Testes de Usabilidade

5.4. Testes de Performance

## Encerramento do Projeto

6.1. Entrega Final ao Cliente

6.2. Reunião de Lições Aprendidas

6.3. Arquivamento dos Documentos do Projeto

## Atribuição de Tarefas

Márcio Sobel (Desenvolvimento) — Criação da interface e telas, integração com a API e armazenamento local, e compilação para Android.

Carlos Ramalho (Scrum Master) — Garantir que o time Scrum siga as premissas e a estrutura do framework.

Assegurar que os processos atendam aos objetivos de todos os envolvidos

Organizar a comunicação.

Orientar os membros do time para que apliquem as práticas ágeis.

Dividir o papel de liderança com o Product Owner.

Promover o desenvolvimento contínuo dos membros do time.

José Gabriel (Documentador) — Responsável pela criação Casos de Uso do projeto.

Organizou e documentou os Casos de Uso em forma de tabela.

Gustavo Portela Pachêco — Estrutura as pastas básicas para o repositório , Criação do repositório git, Definição do fluxo de trabalho no repositório ,Definição do tutorial para fluxo de trabalho, lista de comandos para todas as ações git que forem utilizadas durante o desenvolvimento do projeto. Definição de ferramentas, bibliotecas, frameworks e dispositivos para desenvolver o projeto, comandos de instalação e configuração.

Custos e investimentos

RESUMO - FOLHA DE PAGAMENTO		
CUSTO TOTAL MENSAL	R\$	67.441,70 <i>Custo total para a empresa</i>
SALÁRIOS BASE TOTAIS	R\$	45.960,00 <i>Salário base do funcionário</i>
IMPOSTOS TOTAIS A PAGAR	R\$	16.086,00 <i>Valor dos impostos sobre salário</i>
BENEFÍCIOS (VR, VT, SAÚDE)	R\$	4.596,00 <i>Benefícios ao funcionário</i>
APROVISIONAMENTOS	R\$	799,70 <i>Processos, seguros, despesas</i>

NOME	Carlos Alberto Ramalho Bezerra Neto	
CARGO	Scrum Master	
VALOR POR HORA	R\$	76,00 <i>Inserir um valor médio</i>
TOTAL DE HORAS NO MÊS		200 Horas <i>Horas mensais dedicadas</i>
SALÁRIO BASE	R\$	15.200,00 <i>Salário base do funcionário</i>
IMPOSTOS TOTAIS (% MÉDIO)		35% <i>Média dos impostos sobre salário</i>
IMPOSTOS TOTAIS A PAGAR	R\$	5.320,00 <i>Valor dos impostos sobre salário</i>
BENEFÍCIOS (VR, VT, SAÚDE)	R\$	1.520,00 <i>Benefícios ao funcionário</i>
APROVISIONAMENTOS	R\$	264,48 <i>Processos, seguros, despesas</i>
CUSTO TOTAL MENSAL	R\$	22.304,48 <i>Custo total para a empresa</i>

NOME	Gustavo Portela Pacheco	
CARGO	Gerente de configuração	
VALOR POR HORA	R\$	97,00 <i>Inserir um valor médio</i>
TOTAL DE HORAS NO MÊS		120 Horas <i>Horas mensais dedicadas</i>
SALÁRIO BASE	R\$	11.640,00 <i>Salário base do funcionário</i>
IMPOSTOS TOTAIS (% MÉDIO)		35% <i>Média dos impostos sobre salário</i>
IMPOSTOS TOTAIS A PAGAR	R\$	4.074,00 <i>Valor dos impostos sobre salário</i>
BENEFÍCIOS (VR, VT, SAÚDE)	R\$	1.164,00 <i>Benefícios ao funcionário</i>
APROVISIONAMENTOS	R\$	202,54 <i>Processos, seguros, despesas</i>
CUSTO TOTAL MENSAL	R\$	17.080,54 <i>Custo total para a empresa</i>

NOME	José Gabriel	
CARGO	Documentador 1	
VALOR POR HORA	R\$ 64,00	Inserir um valor médio
TOTAL DE HORAS NO MÊS	180 Horas	Horas mensais dedicadas
SALÁRIO BASE	R\$ 11.520,00	Salário base do funcionário
IMPOSTOS TOTAIS (% MÉDIO)	35%	Média dos impostos sobre salário
IMPOSTOS TOTAIS A PAGAR	R\$ 4.032,00	Valor dos impostos sobre salário
BENEFÍCIOS (VR, VT, SAÚDE)	R\$ 1.152,00	Benefícios ao funcionário
APROVISIONAMENTOS	R\$ 200,45	Processos, seguros, despesas
<b>CUSTO TOTAL MENSAL</b>	<b>R\$ 16.904,45</b>	Custo total para a empresa

NOME	Márcio Cavalcanti Sobel	
CARGO	Desenvolvedor 1	
VALOR POR HORA	R\$ 38,00	Inserir um valor médio
TOTAL DE HORAS NO MÊS	200 Horas	Horas mensais dedicadas
SALÁRIO BASE	R\$ 7.600,00	Salário base do funcionário
IMPOSTOS TOTAIS (% MÉDIO)	35%	Média dos impostos sobre salário
IMPOSTOS TOTAIS A PAGAR	R\$ 2.660,00	Valor dos impostos sobre salário
BENEFÍCIOS (VR, VT, SAÚDE)	R\$ 760,00	Benefícios ao funcionário
APROVISIONAMENTOS	R\$ 132,24	Processos, seguros, despesas
<b>CUSTO TOTAL MENSAL</b>	<b>R\$ 11.152,24</b>	Custo total para a empresa

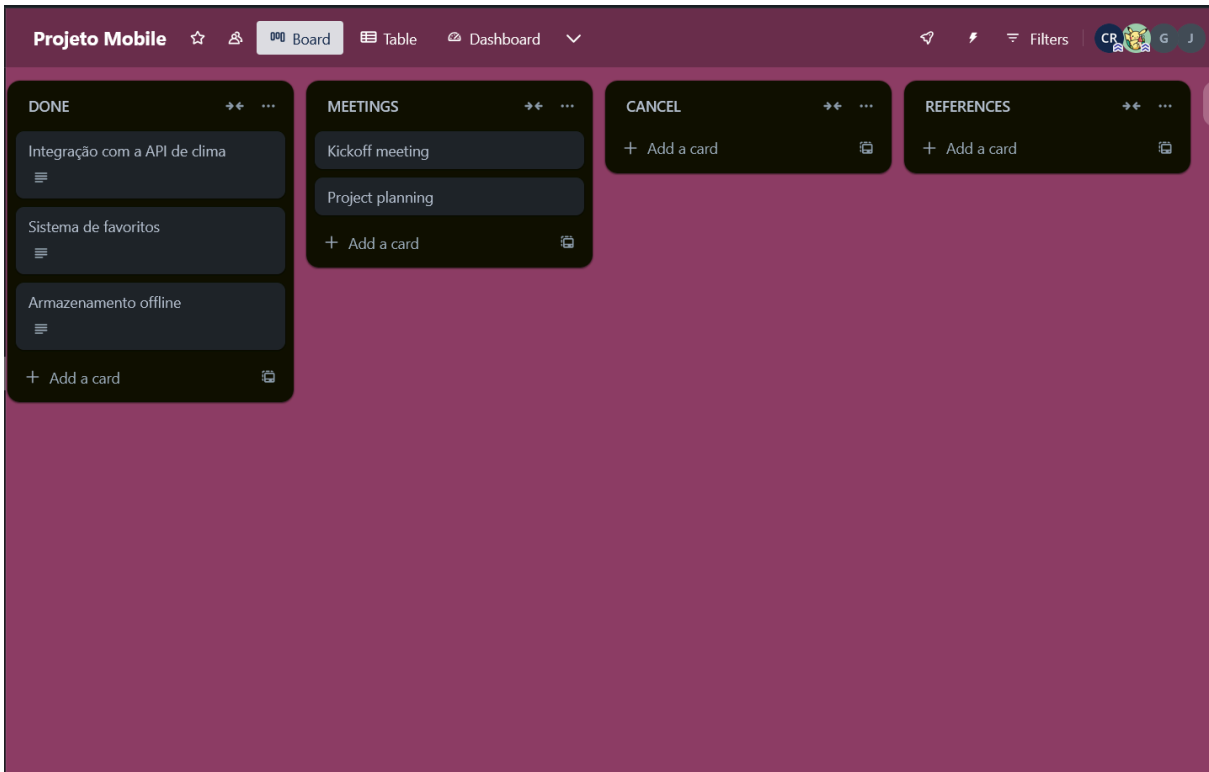
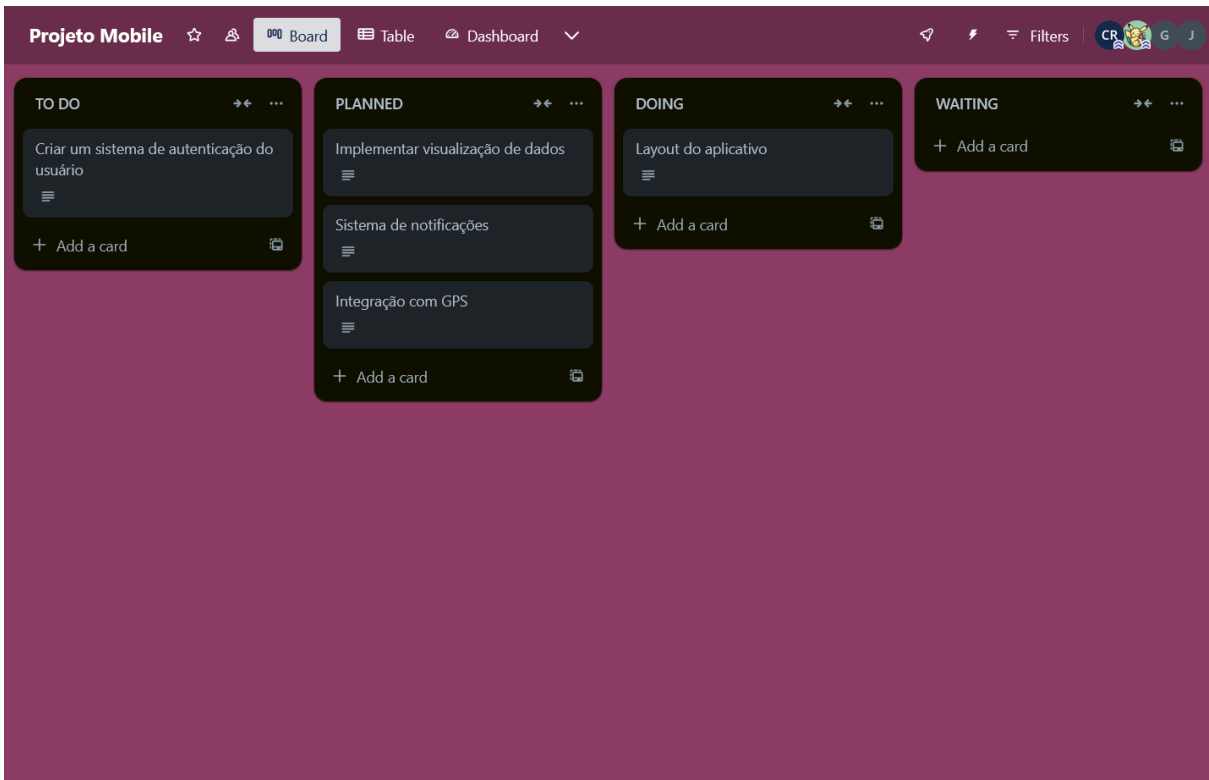


Cronograma com Gantt

TASK	28/10	04/11	11/11	18/11	25/11	02/12/2024(Final)
GitHub				(DONE)		
Documentação						(DONE)
Casos de Uso				(DONE)		
Código do Proj.				(DONE)		
Protótipos			(DONE)			
Apresentação					(DONE)	

# Resumo do Trello

## TRELLO



# Fluxo de trabalho no repositório

Commits on Nov 5, 2024	<div>feat: fetch multiple cities MarcioSobel committed last week</div> <div>ab02eee</div>
Commits on Nov 4, 2024	<div>feat: favorite weathers MarcioSobel committed last week</div> <div>3edaa25</div>
	<div>refactor: change to open weather map api MarcioSobel committed 2 weeks ago</div> <div>9169c9f</div>
	<div>feat: add open meteo api integration MarcioSobel committed 2 weeks ago</div> <div>b554129</div>
Commits on Nov 3, 2024	<div>feat: add base weather card MarcioSobel committed 2 weeks ago</div> <div>8a16507</div>
Commits on Nov 2, 2024	<div>Add project files MarcioSobel committed 2 weeks ago</div> <div>d755626</div>
Commits on Oct 28, 2024	<div>Initial commit gportela02 authored 2 weeks ago</div> <div>Verified802f76e</div>

Commits on Nov 8, 2024	<div>feat: add future temperatures MarcioSobel committed last week</div> <div>c8cf259</div>
	<div>feat: add star icon to weather details MarcioSobel committed last week</div> <div>096bdd3</div>
Commits on Nov 7, 2024	<div>feat: add weather details page MarcioSobel committed last week</div> <div>3b75002</div>
	<div>refactor: refactor caching of weathers MarcioSobel committed last week</div> <div>4ec6505</div>
	<div>feat: add sun to app header MarcioSobel committed last week</div> <div>a7a6fe7</div>
	<div>feat: search locations MarcioSobel committed last week</div> <div>7f44bbc</div>

Commits on Nov 11, 2024	<div>feat: android build MarcioSobel committed 3 days ago</div> <div>5037396</div>
	<div>feat: update README.md MarcioSobel committed 3 days ago</div> <div>f5844eb</div>
Commits on Nov 9, 2024	<div>feat: add future temperatures MarcioSobel committed 5 days ago</div> <div>767d07b</div>
	<div>feat: date selector MarcioSobel committed 5 days ago</div> <div>86d2ddd</div>
	<div>refactor: get temperatures of future days instead of the future 4 hours MarcioSobel committed 5 days ago</div> <div>745fa28</div>

## Casos de Uso

ID	UC001
TÍTULO	Busca de Cidades
ATOR PRINCIPAL	Usuário
OBJETIVO	Permitir que o usuário localize informações de uma cidade específica através de coordenadas.
PRÉ-CONDIÇÃO	<p>O sistema deve estar disponível para busca.</p> <p>O usuário deve acessar o recurso de busca.</p>
PÓS-CONDIÇÃO	<p>O sistema retorna as informações correspondentes à cidade pesquisada.</p> <p>Os resultados da busca são exibidos corretamente na interface para o usuário.</p>
FLUXO PRINCIPAL	<p>O usuário acessa a interface de busca.</p> <p>O sistema exibe um campo de busca com opções de filtro (ex.: nome, estado, país).</p> <p>O usuário insere os critérios de busca e confirma a pesquisa.</p> <p>O sistema processa a busca e retorna uma lista de cidades que atendem aos critérios.</p> <p>O usuário seleciona uma cidade da lista para visualizar mais informações.</p> <p>O sistema exibe os detalhes da cidade selecionada.</p>

FLUXO OPCIONAL	<p>O usuário realiza uma busca inicial.</p> <p>O sistema exibe a lista de resultados.</p> <p>O usuário aplica filtros adicionais (ex.: população, clima, localização).</p> <p>O sistema atualiza a lista com os novos critérios de filtro.</p>
FLUXO ALTERNATIVO	<p>O usuário insere critérios de busca e confirma.</p> <p>O sistema processa a busca e não encontra correspondências.</p> <p>O sistema exibe uma mensagem informando "Nenhuma cidade encontrada" e sugere ajustes nos critérios de busca.</p>
TRATAMENTO DE EXCEÇÕES	<p>O usuário tenta buscar sem inserir critérios.</p> <p>O sistema exibe uma mensagem solicitando ao menos um critério de busca.</p>

ID	UC002
TÍTULO	Gerenciar Favoritos
ATOR PRINCIPAL	Usuário
OBJETIVO	Permitir que o usuário adicione, visualize, remova e gerencie itens em sua lista de favoritos.
PRÉ-CONDIÇÃO	O item que será adicionado/removido aos favoritos deve existir no sistema.
PÓS-CONDIÇÃO	<p>O item é adicionado ou removido com sucesso na lista de favoritos do usuário.</p> <p>O sistema atualiza a lista de favoritos em tempo real.</p>

FLUXO PRINCIPAL	<p>O usuário acessa o item desejado no sistema.</p> <p>O sistema exibe a opção "Adicionar aos Favoritos".</p> <p>O usuário seleciona a opção "Adicionar aos Favoritos".</p> <p>O sistema confirma a inclusão e exibe o item na lista de favoritos.</p>
FLUXO OPCIONAL	<p>O usuário acessa a opção "Favoritos" no menu principal.</p> <p>O sistema exibe a lista de itens marcados como favoritos.</p> <p>O usuário pode interagir com os itens (exibir detalhes, reproduzir, etc.).</p>
FLUXO ALTERNATIVO	<p>O usuário acessa a lista de favoritos.</p> <p>O sistema exibe os itens favoritos com a opção "Remover".</p> <p>O usuário seleciona a opção "Remover" para um item específico.</p> <p>O sistema remove o item da lista e exibe uma mensagem de confirmação.</p>
TRATAMENTO DE EXCEÇÕES	<p>O usuário tenta adicionar um item que já está na lista de favoritos.</p> <p>(O sistema informa que o item já está nos favoritos.)</p>

ID	UC03
TÍTULO	Exibição de Previsão por Dia e Hora
ATOR PRINCIPAL	Usuário

OBJETIVO	<p>Permitir que o usuário visualize a previsão do tempo de uma cidade específica para um dia e hora escolhidos.</p>
PRÉ-CONDIÇÃO	<p>O sistema deve estar conectado a um serviço de previsão do tempo atualizado.</p> <p>O usuário deve ter selecionado uma cidade para consulta.</p>
PÓS-CONDIÇÃO	<p>O sistema exibe a previsão do tempo correspondente ao dia e hora solicitados.</p> <p>O usuário pode visualizar informações detalhadas como temperatura, clima, umidade e vento.</p>
FLUXO PRINCIPAL	<p>O usuário seleciona a cidade desejada.</p> <p>O sistema exibe uma interface com opções de data e hora para consulta.</p> <p>O usuário escolhe o dia e hora para verificar a previsão.</p> <p>O sistema recupera os dados correspondentes à seleção.</p> <p>O sistema exibe as informações da previsão para o dia e hora especificados, incluindo: Temperatura, Condição climática (ex.: ensolarado, nublado, chuva) e Umidade.</p>
FLUXO OPCIONAL	<p>O usuário visualiza a previsão para um horário específico.</p> <p>O sistema exibe a opção de mudar o intervalo (ex.: de hora em hora ou por períodos como manhã, tarde, noite).</p> <p>O usuário escolhe o novo intervalo.</p> <p>O sistema atualiza a previsão exibida conforme o intervalo selecionado.</p>

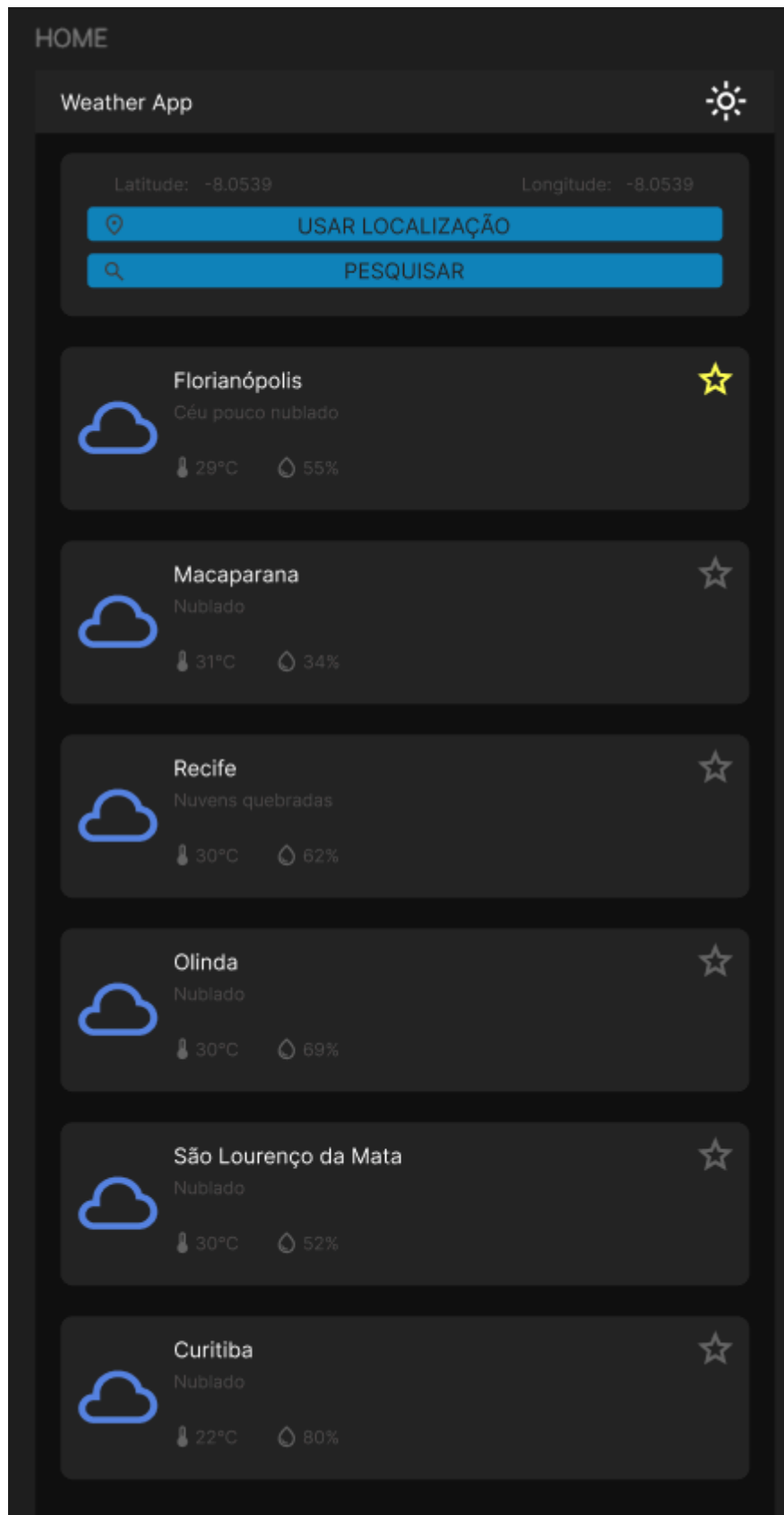
FLUXO ALTERNATIVO	<p>O usuário seleciona um horário específico para consulta.</p> <p>O sistema verifica a disponibilidade dos dados e não encontra previsões para o horário escolhido.</p> <p>O sistema exibe uma mensagem: "Previsão indisponível para o horário selecionado. Por favor, escolha outro intervalo."</p>
TRATAMENTO DE EXCEÇÕES	<p>O usuário tenta consultar a previsão sem selecionar um horário válido.</p> <p>O sistema exibe uma mensagem: "Por favor, selecione um dia e horário para exibir a previsão."</p>

ID	UC04
TÍTULO	Localização Atual via GPS
ATOR PRINCIPAL	Usuário
OBJETIVO	Permitir que o usuário obtenha sua localização geográfica atual utilizando o GPS do dispositivo.
PRÉ-CONDIÇÃO	<p>O usuário deve ter um dispositivo com GPS habilitado.</p> <p>O GPS do dispositivo deve estar ativado e funcionando corretamente.</p>
PÓS-CONDIÇÃO	<p>O sistema obtém e exibe a localização atual do usuário em formato de coordenadas (latitude e longitude) ou como nome de local.</p> <p>O usuário pode visualizar sua localização em um mapa ou usar os dados para outras funcionalidades (como buscar informações ou rotas).</p>

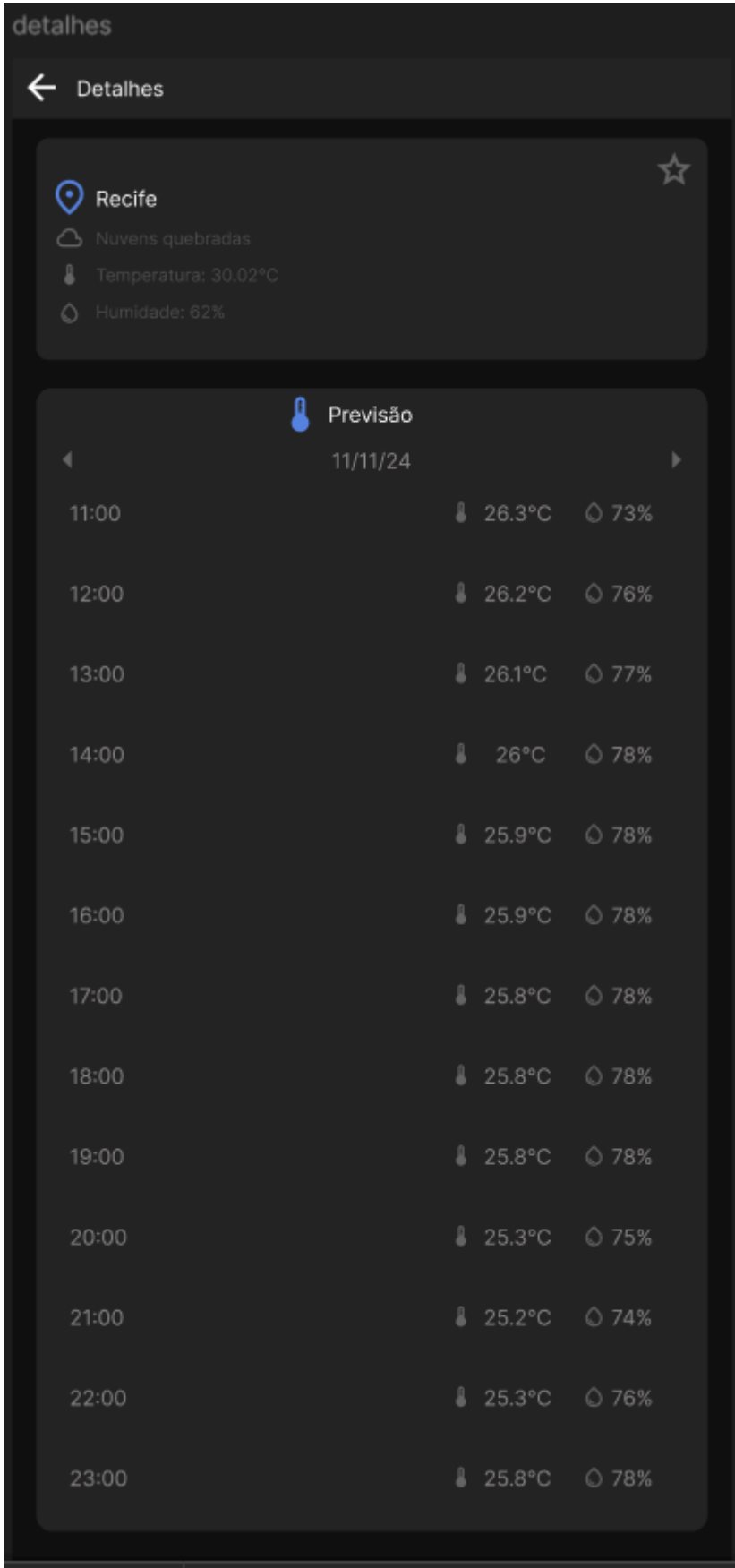


FLUXO PRINCIPAL	<p>O sistema ativa o GPS e começa a buscar a localização do usuário.</p> <p>O sistema recupera a localização atual (latitude e longitude) do usuário.</p> <p>O sistema exibe a localização em um formato legível para o usuário (ex.: nome de cidade, coordenadas, ou em um mapa).</p> <p>O sistema pode oferecer opções adicionais como salvar a localização ou utilizar os dados para outra funcionalidade (ex.: rota).</p>
FLUXO OPCIONAL	<p>O usuário acessa a funcionalidade de "Localização Atual".</p> <p>O sistema exibe a localização atual em um mapa interativo (utilizando Google Maps, OpenStreetMap, etc.).</p>
FLUXO ALTERNATIVO	<p>O usuário acessa a funcionalidade de "Localização Atual".</p> <p>O sistema não consegue obter a localização devido a problemas com o GPS (ex.: sinal fraco ou desativado).</p> <p>O sistema exibe uma mensagem: "Não foi possível determinar sua localização. Verifique se o GPS está ativado e tente novamente."</p>
TRATAMENTO DE EXCEÇÕES	<p>O GPS não está ativado ou não pode ser acessado.</p> <p>O sistema exibe uma mensagem informando ao usuário que o GPS está desativado e solicita a ativação do serviço.</p>

Protótipos  
Página Home (Figma)



Tela de detalhes do Clima e previsão (Figma)



## Extratos do código

(Serviço de armazenamento local)

```
1 import { Injectable } from '@angular/core';
2 import { OpenWeatherMapResponse } from './open-weather-map.service';
3
4 @Injectable({
5   providedIn: 'root'
6 })
7 export class LocalStorageService {
8
9   constructor() { }
10
11   saveWeather(weather: OpenWeatherMapResponse, cacheKey: string = "weathers"): void {
12     weather.favorited = true;
13     const cachedWeathers = localStorage.getItem(cacheKey) || "[]";
14     const weathers = JSON.parse(cachedWeathers) as OpenWeatherMapResponse[];
15     weathers.push(weather);
16     localStorage.setItem(cacheKey, JSON.stringify(weathers));
17   }
18
19   removeWeather(weather: OpenWeatherMapResponse, cacheKey: string = "weathers"): void {
20     weather.favorited = false;
21     const cachedWeathers = localStorage.getItem(cacheKey) || "[]";
22     const weathers = JSON.parse(cachedWeathers) as OpenWeatherMapResponse[];
23     const filteredWeathers = weathers.filter(w => w.name !== weather.name);
24
25     if (filteredWeathers.length === 0) {
26       localStorage.removeItem(cacheKey);
27       return;
28     }
29
30     localStorage.setItem(cacheKey, JSON.stringify(filteredWeathers));
31   }
32
33   getWeathersInCache(cacheKey: string = "weathers"): OpenWeatherMapResponse[] {
34     const cachedWeathers = localStorage.getItem(cacheKey) || "[]";
35     return JSON.parse(cachedWeathers) as OpenWeatherMapResponse[];
36   }
37 }
38
```

(html do componente da página principal)

```
1 <ion-header [translucent]="true">
2   <ion-toolbar>
3     <div>
4       <ion-title>
5         Weather App
6       </ion-title>
7
8       <ion-icon name="sunny-outline" id="header-icon"></ion-icon>
9     </div>
10  </ion-toolbar>
11 </ion-header>
12
13 <ion-content [fullscreen]="true">
14   <ion-card>
15     <ion-card-header>
16       <div>
17         <ion-input label="Latitude:" placeholder="Inserir..." [value]="latitude" (ionChange)="setLatitude($event.target.value)"></ion-input>
18         <ion-input label="Longitude:" placeholder="Inserir..." [value]="longitude" (ionChange)="setLongitude($event.target.value)"></ion-input>
19       </div>
20       <ion-button expand="block" (click)="setLocationWithGPS()">
21         <ion-icon name="location-outline"></ion-icon>
22         Usar localização
23       </ion-button>
24       <ion-button expand="block" (click)="searchWeathers()">
25         <ion-icon name="search-outline"></ion-icon>
26         Pesquisar
27       </ion-button>
28     </ion-card-header>
29   </ion-card>
30
31   <div *ngFor="let weather of weathers">
32     <app-weather-card [weather]="weather" />
33   </div>
34 </ion-content>
```

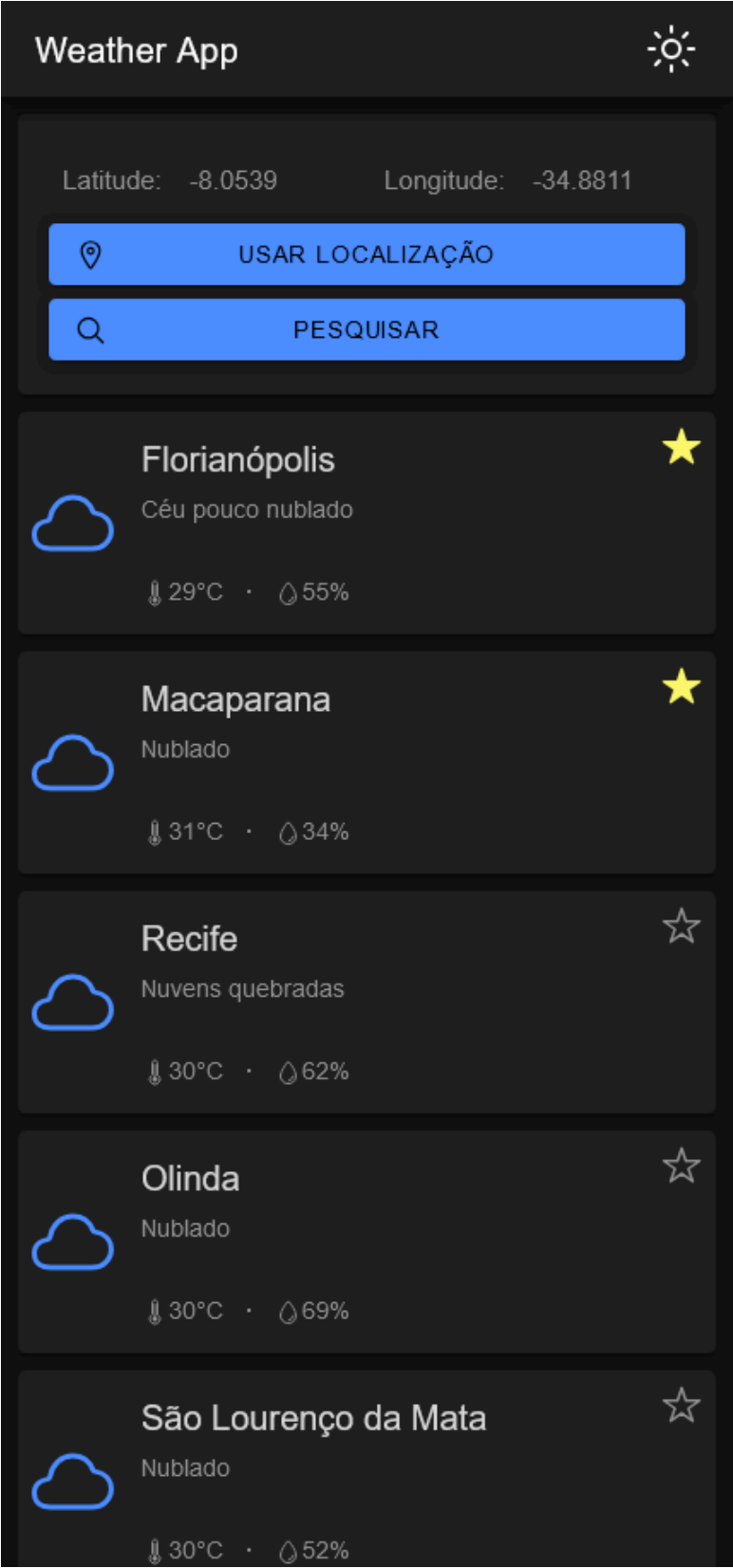
(Código da página principal)

```
1 import { Component, OnInit } from '@angular/core';
2 import { OpenWeatherMapResponse, OpenWeatherMapService } from '../services/open-weather-map.service';
3 import { LocalStorageService } from '../services/local-storage.service';
4 import { Geolocation } from '@capacitor/geolocation';
5
6 @Component({
7   selector: 'app-home',
8   templateUrl: 'home.page.html',
9   styleUrls: ['home.page.scss'],
10 })
11 export class HomePage implements OnInit {
12
13   weathers: OpenWeatherMapResponse[] = [];
14   defaultLatitude: number = -8.0539;
15   defaultLongitude: number = -34.8811;
16
17   latitude: number = this.defaultLatitude;
18   longitude: number = this.defaultLongitude;
19
20   constructor(
21     private openWeatherMapService: OpenWeatherMapService,
22     private localStorage: LocalStorageService
23   ) { }
24
25   ngOnInit() {
26     this.searchWeathers();
27   }
28
29   searchWeathers(latitude: number = this.latitude, longitude: number = this.longitude) {
30     const weathersInCache = this.localStorage.getWeathersInCache();
31     this.weathers = [...weathersInCache];
32
33     weathersInCache.forEach((weather, i) => {
34       this.openWeatherMapService.getCity(weather.coord.lat, weather.coord.lon).subscribe((data) => {
35         data.favorited = true;
36         this.weathers[i] = data;
37       });
38     });
39
40     this.openWeatherMapService.getSurroundingCities(latitude, longitude, 8.1).subscribe(data => {
41       data.forEach((weather) => {
42         const cityNames = this.weathers.map(w => w.name);
43         const alreadyFetched = cityNames.includes(weather.name);
44         if (alreadyFetched) return;
45
46         weather.favorited = false;
47         this.weathers.push(weather);
48       });
49     });
50   }
51
52   setLatitude(input: string | number | null = null) {
53     if (typeof input === "string") {
54       input = parseFloat(input);
55       if (Number.isNaN(input)) return;
56     }
57     if (input === null) {
58       input = this.defaultLatitude;
59     }
60
61     this.latitude = input;
62   }
63
64   setLongitude(input: string | number | null = null) {
65     if (typeof input === "string") {
66       input = parseFloat(input);
67       if (Number.isNaN(input)) return;
68     }
69     if (input === null) {
70       input = this.defaultLongitude;
71     }
72
73     this.longitude = input;
74   }
75
76   async setLocationWithGPS() {
77     const coordinates = await Geolocation.getCurrentPosition();
78     this.setLatitude(coordinates.coords.latitude);
79     this.setLongitude(coordinates.coords.longitude);
80     this.searchWeathers();
81   }
82 }
83
```

(Código que acessa a API de climas)

```
1 import { HttpClient } from '@angular/common/http';
2 import { Injectable } from '@angular/core';
3 import { forkJoin, Observable } from 'rxjs';
4
5 @Injectable({
6   providedIn: 'root'
7 })
8 export class OpenWeatherMapService {
9
10   private url = "https://api.openweathermap.org/data/2.5";
11   private API_KEY = "a";
12
13   constructor(private httpClient: HttpClient) { }
14
15   getCity(latitude: number, longitude: number): Observable<OpenWeatherMapResponse> {
16     const params = new URLSearchParams({
17       lat: latitude.toString(),
18       lon: longitude.toString(),
19       lang: "pt",
20       units: "metric",
21       appid: this.API_KEY,
22     })
23
24     return this.httpClient.get<OpenWeatherMapResponse>(this.url + "/weather?" + params.toString())
25   }
26
27   getSurroundingCities(latitude: number, longitude: number, delta: number = 0.5): Observable<OpenWeatherMapResponse[]> {
28     const coordinates: { lat: number, lon: number }[] = [{
29       lat: latitude,
30       lon: longitude,
31     }, {
32       lat: latitude + delta,
33       lon: longitude + delta
34     }, {
35       lat: latitude - delta,
36       lon: longitude + delta
37     }, {
38       lat: latitude + delta,
39       lon: longitude - delta
40     }, {
41       lat: latitude - delta,
42       lon: longitude - delta
43     }];
44
45     const requests = coordinates.map(
46       ({ lat, lon }) => this.getCity(lat, lon)
47     );
48
49     return forkJoin(requests)
50   }
51 }
```

Demonstração de uso







Recife



Nuvens quebradas



Temperatura: 30.02°C



Humidade: 62%



### Previsão



11/11/24



11:00



26.3°C ·



73%

12:00



26.2°C ·



76%

13:00



26.1°C ·



77%

14:00



26°C ·



78%

15:00



25.9°C ·



78%

16:00



25.9°C ·



78%

17:00



25.8°C ·



78%

18:00



25.8°C ·



78%

19:00



25.8°C ·



75%

20:00



26.3°C ·



72%

## Lições aprendidas

O desenvolvimento de um aplicativo de previsão do tempo com Ionic foi uma experiência rica em aprendizado. Durante o projeto, tivemos a oportunidade de nos aprofundar no uso do Angular, no versionamento de código com Git e nas funcionalidades do Ionic. Também trabalhamos com o Android Studio para testes e depuração, o que expandiu nosso conhecimento sobre plataformas móveis. Além das habilidades técnicas, o trabalho em equipe foi essencial para o sucesso, aprimorando a comunicação e a divisão de tarefas. Planejamento com gráficos de Gantt ajudou a organizar as etapas do projeto e garantir entregas no prazo. Foi um projeto que combinou aprendizado técnico com boas práticas de organização e colaboração.

## Agradecimentos

Gostaríamos de expressar nossa sincera gratidão à Uninassau por proporcionar o suporte e os recursos necessários para a realização deste projeto. Agradecemos especialmente ao professor João Ferreira da Silva Junior por sua orientação valiosa, contribuições e dedicação ao longo do desenvolvimento. Sua experiência e apoio foram fundamentais para o nosso aprendizado e para o sucesso do projeto. Este trabalho não seria possível sem o incentivo e a estrutura oferecidos. Nosso muito obrigado!

## Referências

Angular docs:

<https://v17.angular.io/docs>

Ionic docs:

<https://ionic.io/docs>

Weather API:

<https://openweathermap.org/api>

OpenMeteo:

<https://open-meteo.com/en/docs/elevation-api>