

SIGATC Weather App - AI Development Instructions

Complete Step-by-Step Guide for Building the Prototype

Project Overview

Project Name: SIGATC (Sistema Inteligente de Gestão e Alerta de Tempo Crítico)

Objective: Build a functional front-end weather application for Goiânia, Goiás, Brazil that displays an interactive map with color-coded streets based on flood risk levels and real-time weather forecast data.

Target Audience: General public - anyone who needs accessible weather and flood risk information

Technology Stack: HTML5, CSS3, JavaScript, Leaflet.js (OpenStreetMap), responsive design

Section 1: Application Requirements

1.1 Core Features

The weather app must include:

1. Interactive Map Display

- Centered on Goiânia, Goiás (-16.6869, -49.2648)
- Zoom levels: minimum 11, maximum 18, default 12
- Street-level detail showing major roads and neighborhoods
- Color-coded street overlays (red, yellow, green) based on flood risk

2. Risk Zone Visualization

- **Red Zones (High Risk):** Streets with severe recurring flooding
- **Yellow Zones (Moderate Risk):** Streets with occasional flooding
- **Green Zones (Low Risk):** Streets with minimal flood risk
- Each zone displays name, description, and risk level on hover/click

3. Current Weather Dashboard

- Temperature (Celsius)
- Precipitation (mm)
- Humidity (%)
- Wind speed (km/h)

- Overall risk level indicator with color coding

4. 24-Hour Forecast Panel

- Hourly forecast display
- Temperature trends
- Precipitation probability
- Risk level changes over time

5. User Interface Elements

- Clean, accessible icons for weather conditions
- Mobile-responsive design
- Color-blind friendly palette
- Simple navigation without overwhelming information

Section 2: Data Structure

2.1 Risk Zones Data

Red Zones (8 locations):

1. Marginal Botafogo (Jardim Goiás e Centro) - Severe recurring flooding
2. Avenida Feira de Santana (Parque Amazônia) - Córrego Serrinha overflow
3. Avenida 87 (Setor Sul) - History of blocked roads
4. Vila Redenção (Rua Nonato Mota, Rua C-183) - Córrego Botafogo flooding
5. Residencial Goiânia Viva (Rua VF-82, VF-96) - Recurring flash floods
6. Pedro Ludovico - Intense flooding
7. Parque Industrial João Braz - High risk area
8. Setor Finsocial - Chronic flooding and property damage

Yellow Zones (7 locations):

1. Centro - Av. Paranaíba with Rua 23 (Estádio Olímpico)
2. Ginásio Rio Vermelho
3. Av. Dona Gercina Borges
4. Alameda dos Buritis
5. Bairro Feliz
6. Avenida José Rodrigues de Moraes Neto (Parque Amazonas)
7. Avenida Anhanguera (Lago das Rosas area)

Green Zones (5 locations):

1. North Region (Região Norte)

2. East Region (Região Leste)
3. Cidade Jardim
4. Mendanha
5. Southwest Sector (Setor Sudoeste)

2.2 Weather Data Structure

The application uses the SIGATC dataset which includes:

Meteorological Variables:

- Temperature at 2m (temperature_2m)
- Relative humidity (relative_humidity_2m)
- Dew point (dew_point_2m)
- Wind speed at 10m (wind_speed_10m)
- Wind gusts (wind_gusts_10m)
- Precipitation (precipitation)
- Soil moisture (soil_moisture_0_to_1cm, soil_moisture_1_to_3cm)
- Visibility (visibility)

Derived Risk Indices:

- **Storm Index:** Measures storm potential based on pressure, humidity, temperature differentials
- **SSI (Soil Saturation Index):** Estimates ground saturation and drainage capacity
- **IFI (Immediate Flood Index):** Calibrated indicator for flash flood risk
- **SIGATC Risk Score:** Combined score from 0.0 to 1.0
- **Risk Label:** green (low), yellow (moderate), red (high)
- **Risk Description:** Human-readable explanation in Portuguese

Section 3: AI Prompting Instructions

3.1 Initial Setup Prompt

Use this exact prompt with your AI coding assistant:

```
Create a weather application for Goiânia, Brazil with the following specifications:
```

PROJECT NAME: SIGATC Weather App

MAIN TECHNOLOGY: HTML, CSS, JavaScript with Leaflet.js for mapping

CORE REQUIREMENTS:

1. Display an interactive map of Goiânia centered at coordinates -16.6869, -49.2648
2. Use OpenStreetMap tiles via Leaflet.js (no API key needed)
3. Show colored street overlays for flood risk zones:

- RED for high risk areas
 - YELLOW for moderate risk areas
 - GREEN for low risk areas
4. Display current weather conditions in a dashboard
 5. Show 24-hour forecast with hourly data
 6. Make it fully responsive for mobile and desktop
 7. Use easy-to-understand weather icons
 8. Design for accessibility (color-blind friendly, high contrast)

The application must be a single-page application (SPA) that can run completely in the browser.

3.2 Map Implementation Prompt

After the initial setup, provide this detailed map specification:

STEP 1: MAP CONFIGURATION

Initialize a Leaflet map with these exact settings:

- Container: div with id="map"
- Center coordinates: [-16.6869, -49.2648]
- Initial zoom level: 12
- Min zoom: 11
- Max zoom: 18
- Tile layer: OpenStreetMap standard tiles
- Attribution: OpenStreetMap contributors

Add zoom controls in the top-right corner.

Add a scale control in the bottom-left corner.

Enable touch interactions for mobile devices.

STEP 2: RISK ZONE MARKERS

Create three types of markers with different colors:

RED MARKERS (High Risk Zones):

Use bright red circles (color: #DC2626) with:

- Radius: 300 meters
- Fill opacity: 0.3
- Stroke weight: 2
- Stroke color: #991B1B

Add these 8 red zone locations:

1. Marginal Botafogo at -16.7103, -49.2539
2. Av. Feira de Santana at -16.6731, -49.2394
3. Avenida 87 at -16.7072, -49.2706
4. Vila Redenção at -16.7289, -49.2656
5. Residencial Goiânia Viva at -16.6569, -49.3031
6. Pedro Ludovico at -16.7267, -49.2967
7. Parque Industrial João Braz at -16.7439, -49.3111
8. Setor Finsocial at -16.6872, -49.3097

YELLOW MARKERS (Moderate Risk Zones):

Use amber/yellow circles (color: #F59E0B) with:

- Radius: 250 meters
- Fill opacity: 0.25

- Stroke weight: 2
- Stroke color: #D97706

Add these 7 yellow zone locations:

1. Centro - Av. Paranaiba at -16.6789, -49.2553
2. Ginásio Rio Vermelho at -16.6992, -49.2650
3. Av. Dona Gercina Borges at -16.6900, -49.2789
4. Alameda dos Buritis at -16.6911, -49.2656
5. Bairro Feliz at -16.7011, -49.2817
6. Av. José Rodrigues at -16.6722, -49.2392
7. Av. Anhanguera at -16.7056, -49.2686

GREEN MARKERS (Low Risk Zones):

Use green circles (color: #10B981) with:

- Radius: 400 meters
- Fill opacity: 0.2
- Stroke weight: 2
- Stroke color: #059669

Add these 5 green zone locations:

1. Região Norte at -16.6450, -49.2700
2. Região Leste at -16.6800, -49.2200
3. Cidade Jardim at -16.7200, -49.2400
4. Mendanha at -16.7400, -49.2800
5. Setor Sudoeste at -16.7500, -49.2900

STEP 3: INTERACTIVE POPUPS

For each marker, create a popup that displays:

- Zone name (bold, larger font)
- Risk level with colored badge
- Description of the risk
- Icon indicating the risk type (water drop for flood zones)

Example popup content for red zones:

```
"<strong>Marginal Botafogo</strong><br>
<span>■ RISCO ALTO</span><br>
Alagamentos severos recorrentes"
```

3.3 Weather Dashboard Prompt

Provide these specifications for the weather interface:

WEATHER DASHBOARD DESIGN

Create a modern dashboard with these sections:

SECTION 1: CURRENT CONDITIONS CARD

Position: Top left of the screen, overlaying the map

Style: White background, rounded corners, subtle shadow

Display:

- Large temperature value (e.g., "21.5°C")
- Weather icon matching conditions
- Risk level badge (colored: green/yellow/red)
- Description text: "Risco moderado – instabilidade possível"

Additional metrics in smaller text:

- Precipitation: "0.2 mm"
- Humidity: "91%"
- Wind speed: "6.2 km/h"

SECTION 2: RISK INDICATOR

Position: Below current conditions

Style: Prominent colored banner

Content:

- Risk level in Portuguese
- Traffic light style indicator (●○○)
- Brief explanation

SECTION 3: 24-HOUR FORECAST PANEL

Position: Bottom of screen or right sidebar

Style: Horizontal scrolling cards or vertical list

Each hour shows:

- Time (HH:MM format)
- Temperature
- Precipitation amount
- Small risk level indicator
- Weather icon

Use smooth animations when switching between hours.

Highlight the current hour.

ICONS TO USE:

- ☀ Clear/sunny
- ☁ Partly cloudy
- ☁☁ Cloudy
- ☂ Rain
- ☢ Thunderstorm
- ☔ Water drop for precipitation
- ☺ Wind
- ☃ Temperature

COLOR SCHEME:

- Red zones: #DC2626 (high risk)
- Yellow zones: #F59E0B (moderate risk)
- Green zones: #10B981 (low risk)
- Background: #F9FAFB (light gray)
- Text primary: #111827 (dark gray)
- Text secondary: #6B7280 (medium gray)
- Card backgrounds: #FFFFFF (white)

3.4 Responsive Design Prompt

Ensure mobile compatibility with this prompt:

RESPONSIVE DESIGN REQUIREMENTS

MOBILE VIEW (< 768px):

- Stack dashboard cards vertically
- Map takes 60% of viewport height

- Current conditions card overlays map at top
- Forecast panel appears below map
- Hamburger menu for additional options
- Larger touch targets (minimum 44x44px)

TABLET VIEW (768px - 1024px):

- Dashboard on left side (30% width)
- Map on right side (70% width)
- Forecast panel at bottom
- Collapsible sidebar option

DESKTOP VIEW (> 1024px):

- Dashboard on left sidebar (25% width)
- Map in center (60% width)
- Forecast panel on right (15% width)
- All information visible simultaneously

Use CSS Grid or Flexbox for layout.

Test on multiple screen sizes.

Ensure text remains readable at all sizes.

3.5 Data Integration Prompt

For connecting the actual data:

DATA INTEGRATION INSTRUCTIONS

OPTION 1: Embedded JavaScript Data

Create a JavaScript object containing:

```
const weatherData = {
  currentConditions: {
    temperature: 21.5,
    precipitation: 0.0,
    humidity: 91,
    windSpeed: 6.2,
    riskLevel: "yellow",
    riskScore: 0.383,
    description: "Risco moderado – instabilidade possível",
    timestamp: "2025-12-01 02:00"
  },
  forecast: [
    { time: "2025-11-30 07:00", temp: 19.9, precip: 0.0, risk: "green" },
    { time: "2025-11-30 08:00", temp: 19.7, precip: 0.0, risk: "green" },
    // ... add all 20 forecast hours
  ]
};
```

OPTION 2: External JSON File

Load data from 'weather_app_data.json' using fetch API:

```
fetch('weather_app_data.json')
  .then(response => response.json())
  .then(data => {
    initializeApp(data);
```

```
});
```

OPTION 3: Simulated Live Updates

Create a function to simulate data updates every 5 minutes:

- Slightly vary temperature ($\pm 0.5^{\circ}\text{C}$)
- Randomly add light precipitation
- Recalculate risk scores
- Update UI elements smoothly

For the prototype, use static data from the provided dataset.

Show data timestamp to indicate when it was last updated.

3.6 Accessibility Prompt

Make the app accessible to all users:

ACCESSIBILITY REQUIREMENTS

1. COLOR BLINDNESS SUPPORT:

- Use patterns in addition to colors
- Red zones: solid fill + diagonal stripes
- Yellow zones: solid fill + dots pattern
- Green zones: solid fill only
- Add text labels to all risk indicators

2. SCREEN READER SUPPORT:

- Add ARIA labels to all interactive elements
- Use semantic HTML5 tags (header, main, section, article)
- Provide text alternatives for icons
- Announce risk level changes

3. KEYBOARD NAVIGATION:

- All interactive elements accessible via Tab key
- Map zoom controllable with +/- keys
- Forecast scrollable with arrow keys
- Clear focus indicators

4. TEXT READABILITY:

- Minimum font size: 14px for body text
- High contrast ratios (WCAG AA compliant)
- Line height: 1.5 for body text
- Maximum line length: 70 characters

5. LANGUAGE:

- All interface text in Brazilian Portuguese
- Date/time in Brazilian format (DD/MM/YYYY HH:mm)
- Temperature in Celsius
- Clear, simple language (avoid technical jargon)

3.7 Final Polish Prompt

Add these finishing touches:

FINAL ENHANCEMENTS

1. LOADING STATES:

- Show spinner while map loads
- Skeleton screens for dashboard cards
- Smooth fade-in animations

2. ERROR HANDLING:

- Graceful fallback if map tiles fail to load
- User-friendly error messages in Portuguese
- Retry mechanism for failed data loads

3. PERFORMANCE:

- Lazy load forecast data
- Optimize marker rendering (cluster if > 50 markers)
- Minimize JavaScript bundle size
- Cache tiles for faster subsequent loads

4. POLISH:

- Add subtle hover effects on interactive elements
- Smooth transitions between risk levels
- Animated weather icons
- Pulsing effect on high-risk zones

5. TESTING:

- Test on Chrome, Firefox, Safari, Edge
- Test on iOS and Android devices
- Verify all coordinates display correctly
- Check color contrast ratios
- Validate HTML and CSS

6. DOCUMENTATION:

- Add code comments explaining key functions
- Include a README with setup instructions
- List all third-party libraries used
- Provide sample data structure

Section 4: Complete Example Prompt

4.1 All-in-One Master Prompt

If you want to give everything in a single comprehensive prompt:

BUILD A WEATHER APPLICATION FOR GOIÂNIA, BRAZIL

PROJECT: SIGATC (Sistema Inteligente de Gestão e Alerta de Tempo Crítico)

TECH STACK:

- HTML5, CSS3, JavaScript (vanilla or modern framework)
- Leaflet.js v1.9+ for interactive maps
- OpenStreetMap tiles (no API key required)
- Responsive design (mobile-first approach)

CORE FEATURES:

1. INTERACTIVE MAP

- Center: Goiânia, Goiás at coordinates -16.6869, -49.2648
- Zoom: default 12, range 11-18
- Display colored circular zones for flood risk areas:
 - * RED circles (300m radius, #DC2626): 8 high-risk locations
 - * YELLOW circles (250m radius, #F59E0B): 7 moderate-risk locations
 - * GREEN circles (400m radius, #10B981): 5 low-risk locations
- Each zone has clickable popup with name, risk level, description

2. WEATHER DASHBOARD

- Current conditions card showing:
 - * Temperature (°C)
 - * Precipitation (mm)
 - * Humidity (%)
 - * Wind speed (km/h)
 - * Risk level with colored indicator
- 24-hour forecast panel with hourly data
- Clean weather icons (sun, clouds, rain, etc.)

3. RISK ZONES DATA

RED (HIGH RISK):

1. Marginal Botafogo (Jardim Goiás) at -16.7103, -49.2539 - "Alagamentos severos recorrentes"
2. Av. Feira de Santana at -16.6731, -49.2394 - "Transbordamento do Córrego Serrinha"
3. Avenida 87 (Setor Sul) at -16.7072, -49.2706 - "Histórico de vias bloqueadas"
4. Vila Redenção at -16.7289, -49.2656 - "Alagamentos do Córrego Botafogo"
5. Residencial Goiânia Viva at -16.6569, -49.3031 - "Enxurradas recorrentes"
6. Pedro Ludovico at -16.7267, -49.2967 - "Inundação intensa"
7. Parque Industrial João Braz at -16.7439, -49.3111 - "Risco elevado"
8. Setor Finsocial at -16.6872, -49.3097 - "Enchentes crônicas"

YELLOW (MODERATE RISK):

1. Centro - Av. Paranaíba at -16.6789, -49.2553 - "Alagamentos em pancadas rápidas"
2. Ginásio Rio Vermelho at -16.6992, -49.2650 - "Episódios de alagamento"
3. Av. Dona Gercina Borges at -16.6900, -49.2789 - "Enxurradas moderadas"
4. Alameda dos Buritis at -16.6911, -49.2656 - "Incidência média"
5. Bairro Feliz at -16.7011, -49.2817 - "Invasões d'água repetidas"
6. Av. José Rodrigues at -16.6722, -49.2392 - "Situações moderadas"
7. Av. Anhanguera at -16.7056, -49.2686 - "Tráfego prejudicado"

GREEN (LOW RISK):

1. Região Norte at -16.6450, -49.2700 - "Baixo risco"
2. Região Leste at -16.6800, -49.2200 - "Risco mínimo"
3. Cidade Jardim at -16.7200, -49.2400 - "Ocorrência menos frequente"
4. Mendanha at -16.7400, -49.2800 - "Baixa intensidade"
5. Setor Sudoeste at -16.7500, -49.2900 - "Risco mínimo"

4. SAMPLE WEATHER DATA

Use this for the prototype:

Current: 21.5°C, 0.0mm precip, 91% humidity, 6.2 km/h wind, yellow risk

Forecast (next hours):

- 07:00: 19.9°C, green
- 08:00: 19.7°C, green
- 12:00: 25.1°C, green
- 18:00: 29.6°C, green
- 23:00: 23.3°C, yellow
- 00:00: 22.2°C, yellow

5. DESIGN REQUIREMENTS

- Mobile responsive (breakpoints: 768px, 1024px)
- Color-blind friendly (use patterns + colors)
- Brazilian Portuguese interface
- Accessible (WCAG AA standards)
- Modern, clean UI with weather-appropriate colors
- Smooth animations and transitions

6. LAYOUT

Mobile: Stacked (map top, dashboard below)

Tablet: Two columns (dashboard left, map right)

Desktop: Three sections (dashboard left, map center, forecast right)

7. USER EXPERIENCE

- Touch-friendly on mobile
- Hover effects on desktop
- Loading states for all async operations
- Error handling with friendly messages
- Intuitive navigation

Please create a fully functional prototype that:

- Runs entirely in the browser (client-side only)
- Displays the map with all risk zones
- Shows current weather and forecast
- Is responsive across all devices
- Uses clean, accessible design
- Includes smooth animations

Output the complete HTML, CSS, and JavaScript code needed to run this application.
Include inline comments explaining key sections.

Section 5: Testing and Iteration

5.1 Testing Checklist

After receiving the initial version from the AI, test these aspects:

Visual Testing:

- [] Map loads correctly centered on Goiânia
- [] All 20 risk zones appear with correct colors
- [] Popups display complete information

- [] Weather dashboard is readable
- [] Forecast panel shows all hours

Functional Testing:

- [] Zoom in/out works smoothly
- [] Clicking markers opens popups
- [] Forecast data updates correctly
- [] Risk levels display appropriate colors
- [] Timestamps are accurate

Responsive Testing:

- [] Test on iPhone (iOS Safari)
- [] Test on Android device (Chrome)
- [] Test on iPad/tablet
- [] Test on desktop (Chrome, Firefox, Safari)
- [] Check portrait and landscape orientations

Accessibility Testing:

- [] Tab navigation works for all interactive elements
- [] Screen reader announces risk levels
- [] Color contrast meets WCAG standards
- [] Text remains readable when zoomed 200%
- [] All images have alt text

5.2 Iteration Prompts

If you need refinements, use these follow-up prompts:

For Better Visual Design:

Improve the visual design:

- Add a gradient background to the dashboard
- Use Material Design elevation/shadows
- Implement smooth fade-in animations for markers
- Add weather-appropriate background colors (blue for rain, yellow for sun)
- Use custom weather icons instead of emojis

For Enhanced Interactivity:

Add these interactive features:

- Clicking a risk zone focuses and zooms the map to that location
- Hovering over forecast hours highlights corresponding map zones
- Add a legend explaining the color coding

- Include a search box to find specific streets
- Add a "Current Location" button

For Performance:

Optimize performance:

- Lazy load forecast data (only load when panel is visible)
- Use marker clustering for overlapping zones
- Implement tile caching
- Minify CSS and JavaScript
- Add service worker for offline functionality

For Data Updates:

Add functionality to refresh data:

- Create a refresh button in the dashboard
- Show last update timestamp
- Add auto-refresh every 15 minutes
- Display loading indicator during refresh
- Cache previous data while loading new data

Section 6: Deployment Instructions

6.1 Local Testing

To test the prototype locally:

1. Save all files (HTML, CSS, JS, JSON) in a single folder
2. Open a terminal in that folder
3. Run a local server:
 - Python 3: `python -m http.server 8000`
 - Node.js: `npx http-server`
 - VS Code: Use "Live Server" extension
4. Open browser to `http://localhost:8000`

6.2 Web Hosting

To deploy online:

Option 1: GitHub Pages (Free)

1. Create a GitHub repository
2. Upload all files to the repository
3. Go to Settings → Pages

4. Select main branch as source
5. Access at `username.github.io/repository-name`

Option 2: Netlify (Free)

1. Create account at [netlify.com](#)
2. Drag and drop project folder
3. Get instant URL like `project-name.netlify.app`
4. Automatic deployments on file changes

Option 3: Vercel (Free)

1. Install Vercel CLI: `npm i -g vercel`
2. Run `vercel` in project folder
3. Follow prompts
4. Get production URL

6.3 Custom Domain

To use a custom domain (e.g., [sigatc.com.br](#)):

1. Purchase domain from registrar
2. Add domain in hosting platform settings
3. Update DNS records (A record or CNAME)
4. Wait for DNS propagation (up to 48 hours)
5. Enable HTTPS/SSL (usually automatic)

Section 7: Future Enhancements

7.1 Phase 2 Features

Once the prototype is validated:

Backend Integration:

- Connect to Open-Meteo API for real-time data
- Set up automatic data updates every hour
- Store historical data in database
- Implement user authentication

Advanced Features:

- Push notifications for high-risk alerts
- User-reported flooding incidents

- Social media integration
- Multi-language support (English, Spanish)
- Export forecast data as PDF

Data Visualization:

- Historical flooding charts
- Temperature trends graphs
- Precipitation heatmaps
- Comparative analysis tools

7.2 Mobile App Development

Convert to native mobile apps:

React Native Version:

- Reuse most JavaScript logic
- Add GPS location tracking
- Implement local notifications
- Offline data storage

Progressive Web App (PWA):

- Add manifest.json
- Implement service worker
- Enable "Add to Home Screen"
- Offline functionality
- Push notifications

Section 8: Data Source Attribution

8.1 Data Credits

The application uses data from:

1. Open-Meteo API

- Meteorological data source
- Free tier available
- No API key required for basic use
- Attribution: "Weather data by [Open-Meteo.com](#)"

2. OpenStreetMap

- Map tiles and geographic data

- Open-source mapping platform
- Required attribution: "© OpenStreetMap contributors"

3. SIGATC Dataset

- Proprietary risk analysis model
- Based on historical flooding data
- Developed specifically for Goiânia
- Credit: "Risk analysis by SIGATC"

4. Goiânia Civil Defense

- Historical flooding locations
- Risk zone classifications
- Public safety information
- Credit: "Data source: Defesa Civil de Goiânia"

8.2 Legal Compliance

Ensure the application includes:

- Terms of service
- Privacy policy (if collecting user data)
- Data source attributions
- Disclaimer about forecast accuracy
- Contact information
- Last updated date

Section 9: Troubleshooting

9.1 Common Issues and Solutions

Issue: Map doesn't load

- Solution: Check internet connection, verify Leaflet.js CDN link, check browser console for errors

Issue: Markers don't appear

- Solution: Verify coordinate format (latitude, longitude), check if circles are added to map layer, inspect JavaScript console

Issue: Forecast data missing

- Solution: Verify JSON file path, check fetch API permissions, ensure data structure matches expected format

Issue: Mobile display broken

- Solution: Add viewport meta tag, test media queries, check touch event handlers

Issue: Performance slow

- Solution: Reduce marker count, implement clustering, optimize images, minify code

9.2 Browser Compatibility

Minimum browser versions:

- Chrome: 90+
- Firefox: 88+
- Safari: 14+
- Edge: 90+
- Mobile Safari: iOS 14+
- Chrome Mobile: 90+

Use these polyfills if supporting older browsers:

- fetch API polyfill
- Promise polyfill
- CSS Grid fallback

Section 10: Resources and References

10.1 Documentation Links

Leaflet.js:

- Official docs: <https://leafletjs.com/reference.html>
- Tutorials: <https://leafletjs.com/examples.html>
- Plugins: <https://leafletjs.com/plugins.html>

OpenStreetMap:

- Wiki: <https://wiki.openstreetmap.org>
- Tile usage policy: <https://operations.osmfoundation.org/policies/tiles/>
- Nominatim API: <https://nominatim.org/>

Open-Meteo:

- API documentation: <https://open-meteo.com/en/docs>
- Weather variables: https://open-meteo.com/en/docs#weather_variables
- Forecast API: <https://open-meteo.com/en/docs/forecast-api>

10.2 Design Resources

Weather Icons:

- Weather Icons font: <https://erikflowers.github.io/weather-icons/>
- Meteocons: <https://www.alessioatzeni.com/meteocons/>
- Flaticon weather: <https://www.flaticon.com/packs/weather>

Color Palettes:

- Tailwind CSS colors: <https://tailwindcss.com/docs/customizing-colors>
- Coolors generator: <https://coolors.co/>
- Adobe Color: <https://color.adobe.com/>

UI Inspiration:

- Dribbble weather apps: https://dribbble.com/tags/weather_app
- Behance weather projects: <https://www.behance.net/search/projects?search=weather app>
- Awwwards: <https://www.awwwards.com/websites/weather/>

10.3 Learning Resources

JavaScript:

- MDN Web Docs: <https://developer.mozilla.org/>
- [JavaScript.info](https://javascript.info/): <https://javascript.info/>
- FreeCodeCamp: <https://www.freecodecamp.org/>

Responsive Design:

- CSS-Tricks: <https://css-tricks.com/>
- Responsive Design Patterns: <https://responsivedesign.is/patterns/>
- Grid by Example: <https://gridbyexample.com/>

Appendix A: Complete File Structure

```
sigatc-weather-app/
├── index.html          (Main HTML file)
├── css/
│   ├── styles.css      (Main stylesheet)
│   └── responsive.css  (Media queries)
└── js/
    ├── app.js          (Main application logic)
    ├── map.js           (Map initialization and controls)
    ├── weather.js       (Weather data handling)
    └── utils.js         (Helper functions)
└── data/
    └── weather_app_data.json (Risk zones and forecast data)
```

```

    └── risk_zones.json      (Detailed zone information)
    └── assets/
        ├── icons/          (Weather icons)
        └── images/          (Background images, logos)
    └── README.md           (Project documentation)

```

Appendix B: Sample HTML Structure

```

<!DOCTYPE html>
<html lang="pt-BR">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>SIGATC - Sistema de Alerta Climático Goiânia</title>
    <link rel="stylesheet" href="https://unpkg.com/leaflet@1.9.4/dist/leaflet.css" />
    <link rel="stylesheet" href="css/styles.css">
  </head>
  <body>
    <header>
      <h1>SIGATC</h1>
      <p>Sistema Inteligente de Gestão e Alerta de Tempo Crítico</p>
    </header>

    <main>
      <aside id="dashboard">

        </aside>

        <section id="map-container">
          <div></div>
        </section>

        <aside id="forecast-panel">

          </aside>
    </main>

    <footer>

      </footer>

      <script src="https://unpkg.com/leaflet@1.9.4/dist/leaflet.js"></script>
      <script src="js/app.js"></script>
    </body>
  </html>

```

Appendix C: Risk Classification Thresholds

The SIGATC risk model uses these thresholds:

Green (Low Risk):

- SIGATC Score: 0.00 - 0.35
- Precipitation: < 5mm/hour
- Storm Index: < 0.4
- IFI: < 0.15

Yellow (Moderate Risk):

- SIGATC Score: 0.35 - 0.60
- Precipitation: 5-15mm/hour
- Storm Index: 0.4 - 0.7
- IFI: 0.15 - 0.40

Red (High Risk):

- SIGATC Score: > 0.60
- Precipitation: > 15mm/hour
- Storm Index: > 0.7
- IFI: > 0.40

END OF DOCUMENT

This comprehensive guide provides all instructions needed to prompt an AI coding assistant to build the SIGATC weather application prototype. Follow the sections sequentially for best results.

Document Version: 1.0

Last Updated: November 2025

Author: SIGATC Project Team

Contact: For questions about this guide or the SIGATC project