**Metodologia: Utilização de dados públicos, Tratamento e Visualização de Dados meteorológicos com Access e Chart.js**

**1. Coleta e Estrutura dos Dados no Microsoft Access**

Esta etapa foca na utilização e organização inicial dos dados de previsão do tempo dentro do banco de dados Access.

**1.1. Coleta da Fonte de Dados**

* **Definir as Variáveis:** Identifique quais dados de previsão do tempo serão coletados (ex.: **Data/Hora**, **Temperatura Máxima**, **Temperatura Mínima**, **Precipitação**, **Umidade**, **Velocidade do Vento**, **Direção do vento**).
* **Mecanismo de Coleta:** Como o Access não é ideal para scraping ou consumo direto de APIs, a coleta dos dados deve ser feita de forma externa e, posteriormente, importada:
  + **Opção 1 (Manual/Programada):** Obter dados de APIs meteorológicas (ex.: OpenWeatherMap, INMET, Geoaplicada) usando uma linguagem de programação (como **Python** ou **VBA** no próprio Access, ou até mesmo um *script* externo) e salvar em um formato intermediário (ex.: **CSV** ou **Excel**).
  + **Opção 2 (Importação):** Se os dados já estiverem disponíveis em planilhas ou arquivos de texto, utilize o recurso de **Importação** do Access.

**1.2. Modelagem do Banco de Dados (Access)**

* **Criar a Tabela Principal (Exemplo: PrevisaoDiaria):**
  + **ID** (Chave Primária, Numeração Automática)
  + **DataRegistro** (Data/Hora - *Indexado*)
  + **TemperaturaMax** (Número, Decimal)
  + **TemperaturaMin** (Número, Decimal)
  + **Precipitação** (Número, Inteiro)
  + **Umidade** (Número, Inteiro)
* **Criar Consultas para Visualização:** Desenvolva **consultas (Queries)** específicas no Access que formatem os dados exatamente como o Chart.js precisa (ex.: consulta que retorne apenas a data e a temperatura máxima dos últimos 7 dias). Isso simplifica a extração na próxima etapa.

**2. Tratamento e Preparação dos Dados**

O tratamento garante que os dados sejam válidos, consistentes e otimizados para a visualização.

**2.1. Limpeza e Consistência (No Access/VBA)**

* **Validação de Tipos de Dados:** Certificar-se de que os campos numéricos e de data estão no formato correto.
* **Tratamento de Valores Ausentes (Missing Data):** Decidir se valores nulos (NULL) serão:
  + **Removidos:** Excluir registros incompletos (se for um percentual pequeno).
  + **Preenchidos:** Usar a média/mediana para preencher (em dados de séries temporais, pode-se usar a interpolação).
* **Filtro e Agregação:** Use as **consultas do Access** para fazer o tratamento inicial:
  + **Agregação:** Se necessário, calcular médias diárias, semanais ou mensais (ex.: AVG(TemperaturaMin)).
  + **Filtro:** Limitar o período de tempo a ser visualizado (ex.: WHERE DataRegistro >= DateAdd("m", -1, Date()) para o último mês).

**2.2. Ponte de Dados (Exportação para Formato Web)**

O Chart.js é uma biblioteca JavaScript que roda em um navegador. Ele não pode ler diretamente um arquivo .accdb do Access. É necessária uma etapa intermediária, geralmente um **script do lado do servidor**, para extrair os dados e formatá-los.

* **Tecnologia de Ponte:** Utilize uma linguagem de *backend* (como **PHP**, **Python/Flask/Django** ou **Node.js**) que tenha conectividade com o Access via **ODBC/OLEDB**.
* **Processo de Extração:**
  1. O *script* de *backend* se conecta ao arquivo .accdb (Banco de Dados Access).
  2. Executa a **Consulta (Query)** criada na Etapa 1.2.
  3. Formata o resultado da consulta em **JSON (JavaScript Object Notation)**, que é o formato ideal para consumo por aplicações web (incluindo o Chart.js).

**Estrutura JSON de Exemplo:**

JSON

{

"labels": ["01/Set", "02/Set", "03/Set", ...],

"dataMax": [28, 30, 27, ...],

"dataMin": [18, 20, 16, ...]

}

**3. Visualização e Criação de Gráficos com Chart.js**

Esta etapa utiliza o JSON gerado na etapa 2.2 para criar a interface gráfica de visualização.

**3.1. Estrutura HTML/JavaScript**

* **HTML:** Crie um arquivo HTML contendo o elemento <canvas> que o Chart.js usará para desenhar o gráfico.

HTML

<canvas id="graficoPrevisao"></canvas>

<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>

<script src="app.js"></script>

* **JavaScript (app.js):**
  1. **Requisição de Dados:** Use fetch (ou jQuery AJAX) para chamar o *script* de *backend* (ponto de extremidade/API) que retorna o JSON de previsão do tempo.
  2. **Conversão:** Parsear a resposta JSON.
  3. **Configuração do Chart.js:**
     + Definir o tipo de gráfico mais adequado (ex.: **Linha** para Temperatura e **Barra** para Chuva).
     + Configurar os **rótulos (labels)** com as datas (labels do JSON).
     + Configurar os **datasets** com os dados de temperatura/chuva (dataMax, dataMin etc. do JSON).
     + Para visualização de diferentes escalas (ex.: temperatura em ∘C e chuva em mm), configure **eixos Y múltiplos**.

**3.2. Tipos de Gráficos Sugeridos**

| Dados | Tipo de Gráfico (Chart.js) | Objetivo |
| --- | --- | --- |
| **Temperatura Máx/Min ao longo do tempo** | Linha (line) | Mostrar a tendência e a variação diária. |
| **Precipitação** | Barra (bar) | Visualizar o volume ou a chance de precipitação por dia. |
| **Umidade, Vento, Pressão (variáveis secundárias)** | Gráfico de Radar (radar) ou Gráfico Combinado | Comparar o estado de várias variáveis em um ponto no tempo. |

Exportar para as Planilhas

**4. Manutenção e Atualização**

Para que a metodologia funcione de forma contínua, é essencial automatizar a atualização dos dados.

* **Atualização do Access:** Implementar uma rotina (pode ser um *script* agendado no sistema operacional) que:
  1. Coleta novos dados meteorológicos.
  2. Abre o banco Access e importa/anexa os novos registros.
  3. Executa rotinas de tratamento de dados (VBA, se houver).
* **Atualização da Visualização:** A cada carregamento da página web, o Chart.js fará uma nova chamada ao *script* de *backend*, garantindo que os gráficos exibam a informação mais recente disponível no banco de dados Access.

Esta metodologia cria uma arquitetura robusta onde o Access atua como o **repositório de dados confiável**, o *script* de **backend** atua como a **API de dados** e o **Chart.js** atua como a **interface de visualização** moderna e dinâmica.