

WeatherApp

Computação Distribuída Curso Técnico Superior Profissional em Desenvolvimento Web e Multimédia

> João Taveira a28545 Nicolas Silva a27240 Paulo Júnior a27261 Edgar Sousa a28557 Simão Fernandes a28547



Tópicos

- Introdução
- Equipa
- Problema
- Solução Proposta
- Diagrama de Arquitetura
- Partes do Código
- Github
- Conclusão



Introdução

O nosso trabalho é uma aplicação web em que inserimos o nome de uma cidade e consultamos o clima (API) com base na cidade, sugerimos um filme (API).

Exemplo: Braga? Sugerimos um filme com o tema da cidade.



Equipa

João Taveira - Desenvolvimento do Frontend e ajudou no desenvolvimento da API de filmes

Nicolas Silva - Desenvolvimento do Frontend e API do tempo

Paulo Júnior - Configuração da SupaBase

Edgar Sousa - Docker e Configuração do Servidor

Simão Fernandes - Desenvolvimento do backend (API de Filme)

Todos contribuímos para criar uma solução simples e funcional.





Problema

Muitas pessoas têm dificuldade em encontrar informações relevantes sobre o clima de uma cidade específica de forma integrada com opções de lazer, como sugestões de filmes. Por vezes, utilizadores desejam explorar conteúdos (como filmes) relacionados a uma cidade específica, seja por curiosidade, planeamento de viagens ou para aproveitar o momento. No entanto, não há uma ferramenta simples que combine essas duas funcionalidades: consultar o clima de uma cidade e obter sugestões de filmes baseadas na cidade escolhida.



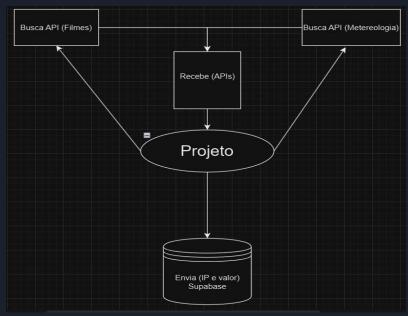
Solução Proposta

Nossa aplicação web resolve esse problema ao permitir que o utilizador insira o nome de uma cidade e, com base nisso, consulte o clima atual usando uma API meteorológica. Além disso, a aplicação sugere filmes que têm conexão com a cidade inserida, utilizando uma API de filmes. Essa solução combina praticidade e entretenimento, integrando informações úteis com uma experiência de lazer personalizada, tudo em uma única plataforma.



Diagrama de Arquitetura

O diagrama de arquitetura representa um projeto que integra diferentes APIs e armazena informações em um banco de dados Supabase. O sistema busca dados de duas APIs externas: uma relacionada a filmes e outra a meteorologia.





Partes do Código - Service Filmes

API de filmes código para dar o respetivo título, data, pôster e avaliação (Front-End)

```
const apiKeyTMDB = "1faf5bfaeee634f33cfefcd45fa3fb78";
const apiUrlTMDB = "https://api.themoviedb.org/3/search/movie";
```

```
function shuffleArray(array) {
  const shuffled = array.slice();
  for (let i = shuffled.length - 1; i > 0; i--) {
    const j = Math.floor(Math.random() * (i + 1));
    [shuffled[i], shuffled[j]] = [shuffled[j], shuffled[i]];
  }
  return shuffled;
}
```



Partes do Código - API Tempo / Filmes

server.js

```
server.get("/api/weather", async (reg, res) => {
 const { city } = req.query;
   return res.status(400).json({ error: "Cidade não especificada." });
  try
   const apiUrl = `https://api.openweathermap.org/data/2.5/weather?q=${city}&appid=${API KEY}&units=metric`;
    const response = await fetch(apiUrl);
   const weatherData = await response.json();
   if (!weatherData || weatherData.cod !== 200) {
     return res.status(400).json({ error: "Cidade não encontrada." });
    const movies = await movieService.getMovieSuggestions(city);
    res.json({
     weather: weatherData,
     movies.
    catch (error)
   console.error("Erro ao buscar dados da API:", error);
   res.status(500).json({ error: "Erro ao buscar dados da API" });
```

Partes do Código - Supabase

supabase.js

```
const supabase = createClient(supabaseUrl, supabaseKev);
 const inputValue = document.getElementById("city name").value;
      .insert([{ ip: userIP, input value: inputValue }]);
```



Partes do Código - Supabase

server.js

```
server.get("/src/data/supabase.js", (req, res) => {
  res.sendFile(path.join(__dirname, "src", "data", "supabase.js"));
});
```



Partes do Código - Supabase

dataaccess.js

```
async function capturarDados() {
    const { data, error } = await supabase
     .from("user_data")
      .insert([{ ip: userIP, input_value: inputValue }]);
    if (error) {
     console.error("Erro ao salvar no Supabase:", error);
     alert("Erro ao salvar os dados!");
    } else {
      console.log("Dados salvos com sucesso:", data);
     alert("Dados salvos com sucesso!");
  } catch (error) {
   console.error("Erro ao capturar IP:", error);
   alert("Não foi possível capturar o IP.");
document.getElementById("search").addEventListener("submit", async (event) => {
 event.preventDefault();
  await capturarDados();
```



Partes do Código - Script.js

```
document.guerySelector("#search").addEventListener("submit", async (event) => {
  event.preventDefault();
  const cityName = document.querySelector("#city_name").value.trim();
  if (!cityName) {
   const weatherElement = document.guervSelector("#weather");
   const footerElement = document.guerySelector("#footer");
    if (weatherElement) weatherElement.classList.remove("show");
    if (footerElement) footerElement.classList.remove("show");
   showAlert("Você precisa digitar uma cidade...");
   const response = await fetch(
       /api/weather?city=${encodeURIComponent(cityName)}
   if (!response.ok) {
     throw new Error("Cidade não encontrada ou erro no servidor.");
   const { weather, movies } = await response.json();
    showInfo({
     city: weather.name.
      country: weather.sys.country,
      temp: weather.main.temp,
      tempMax: weather.main.temp max,
      tempMin: weather.main.temp min.
      description: weather.weather[0].description,
      tempIcon: weather.weather[0].icon.
     windSpeed: weather.wind.speed,
     humidity: weather.main.humidity,
    showMovieSuggestions(movies);
```

```
const footerElement = document.guerySelector("#footer");
   if (footerElement) footerElement.classList.add("show");
   const weatherElement = document.guerySelector("#weather");
   if (weatherElement) weatherElement.classList.remove("show");
   if (footerElement) footerElement.classList.remove("show");
   showAlert(
   console.error("Erro:", error);
function toggleHiddenOnFocus() {
 const cityNameInput = document.getElementById("city name");
 cityNameInput.addEventListener("focus", () => {
   submitButton.hidden = true;
 cityNameInput.addEventListener("blur", () => {
   submitButton.hidden = false;
function showInfo(json) {
 let tempElement = document.getElementById("temp");
 switch (true) {
   case json.temp <= 0:
     tempElement.style.backgroundColor = "#0ea5e9";
   case json.temp > 0 && json.temp <= 15:
     break:
   case ison.temp > 15 && ison.temp <= 30:
     tempElement.style.backgroundColor = "#fbbf24":
   case json.temp > 30:
     tempElement.style.backgroundColor = "red";
```

```
document.getElementById("title").innerHTML = "${json.city}, ${json.country}";
 document.getElementById("temp_value").innerHTML = `${json.temp.toFixed(
 document.getElementById("temp_description").innerHTML = "${json.description}";
   .getElementById("temp img")
 document.getElementById("temp_max").innerHTML = ${json.tempMax.toFixed(
 document.getElementById("temp_min").innerHTML = '${
   ison.tempMin.toFixed(0) - 14
 document.getElementById("humidity").innerHTML = "${json.humidity}%";
 document.getElementBvId("wind").innerHTML = '$(ison.windSpeed.toFixed)
function showMovieSuggestions(movies) 
 const movieSuggestionsContainer =
 movieSuggestionsContainer.innerHTML = "":
 if (movies && movies.length > 0) {
   movies.forEach((movie) => {
     const movieItem = document.createElement("div");
      if (movie.poster) {
       movieItem.innerHTML =
           <img src="${movie.poster}" alt="${movie.title}" />
           ${movie.title}
       movieItem.textContent = movie.title;
     movieSuggestionsContainer.appendChild(movieItem);
   document.querySelector("#movie_section").classList.add("show");
   document.querySelector("#movie section").classList.add("show");
```



Partes do Código - Docker Compose

```
services:
  app:
    build:
      context: .
      dockerfile: dockerfile
    ports:
      - "3000:3000"
    volumes:
      - .:/app
      - /app/node modules
   environment:
      - NODE ENV=development
```



Partes do Código - DockerFile

```
FROM node:22
WORKDIR /src
COPY package*.json ./
COPY server.js ./
RUN npm install
COPY . .
EXPOSE 3000
CMD ["npm", "start"]
```

Github

https://github.com/Paulosouzx/Weather



Conclusão

A nossa aplicação junta de forma simples e prática informações sobre o clima e sugestões de filmes, tudo numa plataforma interativa. Ao usar APIs, mostramos como as tecnologias web podem facilitar o dia a dia, ao mesmo tempo que criamos espaço para melhorias no futuro, sempre a pensar em tornar a experiência mais útil e agradável para os utilizadores.