# Curso Ebac - Javascript

**Documento de Arquitetura de Software**

**Versão <1.0>**6

# Documento de Arquitetura de Software

## Introdução

Está sendo desenvolvido no curso de Javascript da EBAC, uma aplicação utilizando HTML, CSS, Javascript, React e Vite para uma tela de meteorologia.

A primeira premissa do software é trazer um card indicando a cidade do usuário, seguido pela temperatura atual.

No segundo container, ou divisão do site, deve conter as informações sobre a temperatura dos próximos cinco dias, indicando a data, temperatura mínima e máxima.

Inicialmente os dados serão fictícios, mas posteriormente, deve estar aberto a receber uma API externa para se conectar à aplicação.

### Finalidade

Esse software será desenvolvido como projeto final do curso, para que o aluno possa aplicar seus conhecimentos de HTML, CSS, JavaScript e bibliotecas no desenvolvimento de uma aplicação final para o usuário.

### Escopo

A aplicação terá como objetivo fornecer informações meteorológicas de maneira clara. Sua funcionalidade principal será apresentar a temperatura atual e a previsão para os próximos cinco dias. Ela será integrada a uma API meteorológica para obter dados em tempo real.

### Definições, Acrônimos e Abreviações

**HTML**: HyperText Markup Language – Linguagem de marcação utilizada para estruturar conteúdos na web.

**CSS**: Cascading Style Sheets – Linguagem utilizada para estilizar elementos HTML.

**JavaScript**: Linguagem de programação utilizada para criar interatividade e funcionalidades dinâmicas na web.

**React**: Biblioteca JavaScript para criar interfaces de usuário baseadas em componentes.

**Vite**: Ferramenta de build que oferece um ambiente rápido para desenvolvimento front-end.

**API**: Application Programming Interface – Conjunto de rotinas e padrões de programação para acesso a um aplicativo ou plataform

### Visão Geral

Este documento descreve os aspectos arquiteturais da aplicação meteorológica, incluindo a representação da arquitetura, metas, restrições e detalhes sobre sua implementação. O objetivo é garantir que todos os envolvidos compreendam os padrões e escolhas arquiteturais adotados no projeto.

### Representação Arquitetural

A arquitetura do sistema será baseada em uma aplicação front-end utilizando React como biblioteca principal. Serão utilizados componentes reutilizáveis e estilização modular com CSS ou styled-components. O Vite será usado para otimizar o processo de desenvolvimento e build.

**Componentes Principais**:

**Card de Localização**: Mostrar a cidade do usuário e a temperatura atual.

**Previsão Extendida**: Apresentar os dados meteorológicos para os próximos cinco dias.

**Tecnologias Utilizadas**:

React

HTML e CSS (ou styled-components)

Vite para configuração do projeto.

## Metas e Restrições da Arquitetura

### Metas

Fornecer uma interface simples e intuitiva para os usuários.

Garantir performance rápida durante o desenvolvimento e a execução.

### Restrições

O projeto será desenvolvido exclusivamente com tecnologias front-end.

A aplicação inicial utilizará dados fictícios, sem conexão com APIs reais.

## Visão de Casos de Uso

*[Esta seção lista casos de uso ou cenários do modelo de casos de uso quando eles representam funcionalidade central e significativa do sistema final ou, quando têm uma grande cobertura arquitetural — eles experimentam muitos elementos arquiteturais ou quando enfatizam ou ilustram um ponto complexo e específico da arquitetura.]*

### Realizações de Casos de Uso

*[Esta seção ilustra o funcionamento do software, apresentando algumas realizações (ou cenários) de casos de uso selecionadas e explica como os diversos elementos do modelo de design contribuem para a respectiva funcionalidade.]*

## Visão Lógica

*[Esta seção descreve as partes significativas do ponto de vista da arquitetura do modelo de design, como sua divisão em subsistemas e pacotes. Além disso, para cada pacote significativo, ela mostra sua divisão em classes e utilitários de classe. Apresente as classes significativas do ponto de vista da arquitetura e descreva suas responsabilidades, bem como alguns relacionamentos, operações e atributos de grande importância.]*

### Visão Geral

*[Esta subseção descreve toda a decomposição do modelo de design em termos de camadas e de hierarquia de pacotes.]*

### Pacotes de Design Significativos do Ponto de Vista da Arquitetura

*[Para cada pacote significativo, inclua uma subseção com o respectivo nome, uma breve descrição e um diagrama com todos os pacotes e classes significativos nele contidos.*

*Para cada classe significativa no pacote, inclua o respectivo nome, uma breve descrição e, opcionalmente, uma descrição de algumas das suas principais responsabilidades, operações e atributos.]*

## Visão de Processos

*[Esta seção descreve a decomposição do sistema em processos leves (threads simples de controle) e processos pesados (agrupamentos de processos leves). Organize a seção em grupos de processos que se comunicam ou interagem. Descreva os modos principais de comunicação entre processos, como transmissão de mensagens e interrupções.]*

## Visão de Implantação

*[Esta seção descreve uma ou mais configurações da rede física (hardware) na qual o software é implantado e executado. Ela é uma visão do Modelo de Implantação. No mínimo, para cada configuração, ela deve indicar os nós físicos (computadores, CPUs) que executam o software e suas interconexões (barramento, LAN, ponto a ponto, etc.) É incluído também um mapeamento dos processos da* ***Visão de Processos*** *nos nós físicos.]*

## Visão da Implementação

*[Esta seção descreve a estrutura geral do modelo de implementação, a divisão do software em camadas e os subsistemas no modelo de implementação e todos os componentes significativos do ponto de vista da arquitetura.]*

### Visão Geral

*[Esta subseção nomeia e define as diversas camadas e o seu conteúdo, as regras que determinam a inclusão em uma camada específica e as fronteiras entre as camadas. Inclua um diagrama de componentes que mostre os relacionamentos entre as camadas. ]*

### Camadas

*[Para cada camada, inclua uma subseção com o respectivo nome, uma lista dos subsistemas localizados na camada e um diagrama de componentes.]*

### Visão de Dados (opcional)

*[Uma descrição da perspectiva de armazenamento de dados persistentes do sistema. Esta seção será opcional se os dados persistentes forem poucos ou inexistentes ou se a conversão entre o Modelo de Design e o Modelo de Dados for trivial.]*

## Tamanho e Desempenho

*[Uma descrição das principais características de dimensionamento do software que têm um impacto na arquitetura, bem como as restrições do desempenho desejado.]*

## Qualidade

*[Uma descrição de como a arquitetura do software contribui para todos os recursos (exceto a funcionalidade) do sistema: extensibilidade, confiabilidade, portabilidade e assim por diante. Se essas características possuírem significado especial, como implicações de segurança, garantia ou privacidade, elas deverão ser delineadas claramente.]*