

---

**Curso Ebac - Javascript**

**Documento de Arquitetura de Software**

**Versão <1.0><sub>6</sub>**

# Documento de Arquitetura de Software

## 1. 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.

*[A introdução do **Documento de Arquitetura de Software** fornece uma visão geral do documento inteiro. Ela inclui a finalidade, o escopo, as definições, os acrônimos, as abreviações, as referências e a visão geral do **Documento de Arquitetura de Software**.]*

### 1.1 Finalidade

Esse software está sendo desenvolvido como projeto final do curso JavaScript na Ebac, com a finalidade de aplicar os conhecimentos de HTML, CSS, JavaScript e bibliotecas, aprendidos durante o curso, no desenvolvimento de uma aplicação final para o usuário.

*[Esta seção define o papel ou finalidade do **Documento de Arquitetura de Software**, na documentação do projeto como um todo, e descreve rapidamente a estrutura do documento. O público-alvo específico do documento é identificado, com uma indicação de como ele espera usar o documento.]*

### 1.2 Escopo

O documento visa estabelecer diretrizes para o desenvolvimento do produto, bem como apresentar a arquitetura pensada para o processo.

A influência que recai sobre ele é sim a finalização do curso mas também a necessidade de acompanhar o clima num mundo em que a crise climática passou a afetar cada vez mais o dia a dia das pessoas comuns.

*[Uma breve descrição da utilidade do Documento de Arquitetura de Software, do que é afetado por esse documento ou influenciado por ele.]*

### 1.3 Definições, Acrônimos e Abreviações

*[Esta subseção contém as definições de todos os termos, acrônimos e abreviações necessários para interpretar corretamente o **Documento de Arquitetura de Software**. Essas informações podem ser fornecidas fazendo referências ao Glossário do projeto.]*

### 1.4 Visão Geral

*[Esta subseção descreve o que o restante do **Documento de Arquitetura de Software** contém e explica como o documento está organizado.]*

## 2. Representação Arquitetural

*[Esta seção descreve qual é a arquitetura de software do sistema atual e como ela é representada. Da **Visão de Casos de Uso**, **Visão Lógica**, **Visão de Processos**, **Visão de Implantação** e **Visão de Implementação**, enumera as visões necessárias e, para cada visão, explica quais tipos de elementos de modelo ela contém.]*

## 3. Metas e Restrições da Arquitetura

*[Esta seção descreve os requisitos e objetivos do software que têm algum impacto sobre a arquitetura; por exemplo,*

*segurança, garantia, privacidade, uso de um produto desenvolvido internamente e pronto para ser usado, portabilidade, distribuição e reutilização. Ela também captura as restrições especiais que podem ser aplicáveis: estratégia de design e implementação, ferramentas de desenvolvimento, estrutura das equipes, cronograma, código-fonte legado e assim por diante.]*

## 4. 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.]*

### 4.1 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.]*

## 5. 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.]*

### 5.1 Visão Geral

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

### 5.2 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.]*

## 6. 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.]*

## 7. 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.]*

## 8. 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.]*

### 8.1 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.]*

### 8.2 Camadas

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

## **9. 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.]*

## **10. 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.]*

## **11. 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.]*