Universidade Federal do Rio Grande
Centro de Ciências Computacionais - C3
Professor: Bruno Lopes Dalmazo

Relatório Técnico: Cliente/Servidor

Bruna dos Santos Freitas - 132780
Andrew Flores Brongar- 116500

Rio Grande
2022

1. Motivação e definição do problema

O presente relatório tem como objetivo apresentar uma aplicação de cliente/servidor para web que consome uma API de consulta de CEP. O desenvolvimento deste trabalho visa agregar mais conhecimento sobre a estrutura cliente-servidor e o gerenciamento do consumo de API`s.

Tendo como escolha diversas API's, escolheu-se a API de busca de CEP de endereçamento do Brasil denominada ViaCep, na qual, ao realizar uma requisição http para a API, obtém-se o retorno com informações como: CEP, nome da cidade, código do município, UF, etc.

Ademais, focando no desenvolvimento por parte do cliente, adicionou-se a opção do usuário poder consultar e visualizar através de um mapa online no site, a sua localização geográfica, através da latitude e longitude do endereço consultado na API ViaCEP.

2. Arquitetura e regras da aplicação

A aplicação do cliente servidor foi dividida em duas partes, utilizando como base a linguagem JavaScript. Para a construção da página da web foi empregado a linguagem de marcação HTML, CSS e JavaScript. O HTML possibilitou a criação e estruturação de seções, parágrafos e links usando elementos, tags e atributos. Com relação ao JavaScript, teve como objetivo simplificar o desenvolvimento de scripts executáveis e dinâmicos integrados à página no HTML.

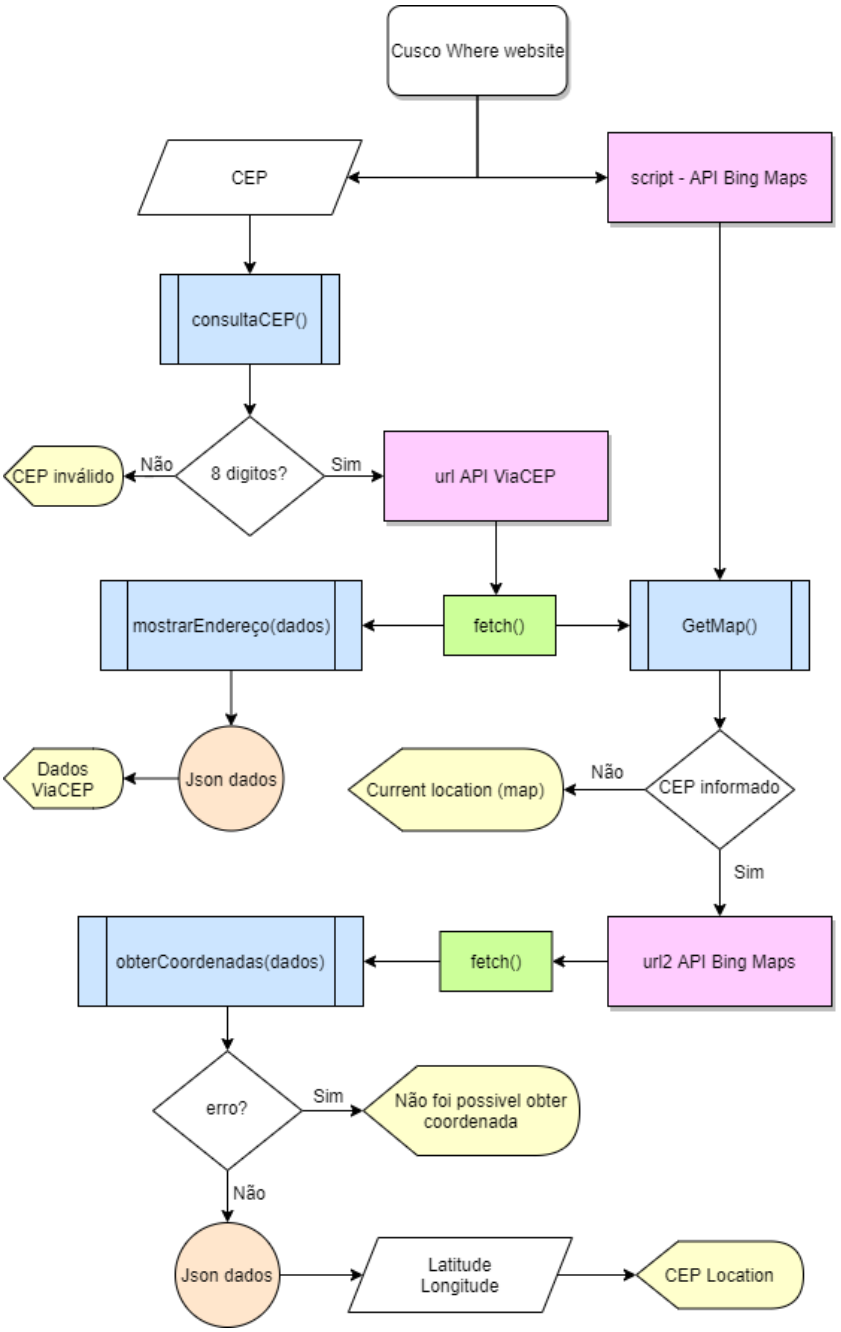
A segunda parte, referente ao consumo do servidor, baseou-se no uso de ferramentas de gerenciamento e consumo de informações do lado do cliente, utilizando a API Fetch, que fornece uma interface JavaScript para acessar e manipular partes do pipeline HTTP. Em suma, a API fetch utiliza o Promisse, que de maneira oculta, tem como retorno um objeto para o qual é possível adicionar callbacks. Logo, com as Promisses, otimiza-se a necessidade do uso de passar callbacks para uma função, tornando a chamada de função assíncrona.

Para obter as informações referentes ao CEP do lado do cliente, utilizou-se a API ViaCEP, um webservice gratuito para consultar códigos de endereçamento postal (CEP). Através de uma requisição http é possível obter o retorno com

informações como, por exemplo, o CEP, nome da cidade, código do município, UF, entre outros serviços.

Para aperfeiçoar a consulta do CEP do lado do cliente, utilizou-se a API Bing Maps da Microsoft, que oferece recursos com os quais é possível mostrar mapas customizados, acessar localizações, endereços e rotas, dentre outras possibilidades. Neste caso de uso, utilizou-se esta API, para consumir informações complementares, como longitude e latitude, a partir do CEP informado pelo cliente.

Figura 1 - Fluxograma de processos



3. Tratamento da saída do server e integração com o frontend

Conforme citado anteriormente, foi utilizada a API fetch para consumir os dados da API ViaCep. No frontend, utilizou-se o método querySelector que retorna o elemento do seletor do arquivo CSS. Tendo o número referente ao CEP a ser fornecido como parâmetro da requisição da API, baseou-se num template string do JavaScript para montar a URL. Após, utilizou-se o método global fetch para realizar a chamada na API e transformar dados em objetos, como resposta, é executada uma função para tratar os objetos e inseri-los, através do innerHTML, na estrutura do frontend como resultado da pesquisa.

Em relação à API do Bing Maps, onde, inicialmente apresenta a posição atual do usuário, a API é solicitada pelo método script do HTML, nele, há um callback de uma função definida no script, logo em seguida, será exibido o mapa na página web. Esse mecanismo também está correlacionado com a API da ViaCEP, uma vez fornecido o CEP para a busca, por um fetch de uma Rest API fornecido pelo Bing Maps, é obtido informações da latitude e longitude. Através dessa interação, torna-se possível a exibição do mapa na localização aproximada do CEP.

4. Dificuldades encontradas

A dificuldade inicial do projeto foi a escolha da temática do WebService a ser escolhida, onde ao analisar escolhemos o consumo de uma API de consulta de CEP. O entendimento de como funcionava a API de mapas foi um dos desafios iniciais, pois a existência e a finalidade geral desta biblioteca já era conhecida, porém, nunca havíamos utilizado ela em alguma aplicação. Sobre a parte de Frontend utilizando HTML, CSS e JavaScript, foi encontrada pouca ou nenhuma dificuldade, pois é uma parte do desenvolvimento que já temos familiaridade.

5. Conclusão

Neste trabalho, desenvolvemos uma aplicação cliente/servidor para web, que usa algumas funcionalidades de um servidor web já existente, no nosso caso de uma API de consulta de CEP`s. Dessa forma, o trabalho foi focado na implementação por parte do cliente, pois o servidor foi acessado via Web Service.

A biblioteca empregada para o consumo das API's foi a Fetch, que como já foi citado anteriormente, fornece uma interface JavaScript para acessar e manipular partes do pipeline HTTP, tais como os pedidos e respostas. Ela também fornece o método global fetch(), utilizado no trabalho, que fornece uma maneira fácil e lógica para buscar recursos de forma assíncrona através da rede.

Por fim, pode-se concluir que, através desse trabalho, foi possível adquirir mais conhecimentos relacionados ao tratamento de dados, estrutura cliente-servidor, o gerenciamento do consumo de API's, e a responsividade do site hospedado para todos os modelos de telas.