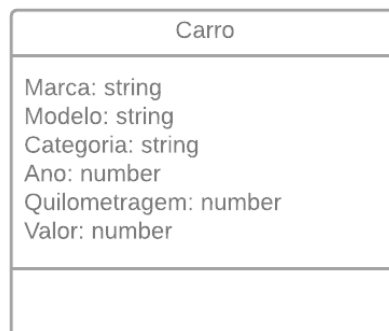


# DO.DEV SCHOOL

## DoDev School

### Lista de exercícios - APIs

1. Criar APIs com um CRUD para a entidade conforme o diagrama usando o DronaHQ, teste as APIs usando o insomnia para garantir que tudo está funcionando.



2. Crie uma classe conforme o diagrama, esta classe não precisa estar com as propriedades encapsuladas porém precisa de um construtor!
3. Depois disso o desafio é implementar as APIs: GET, GET by id, POST, PUT, DELETE no código usando o typescript.

Para isso vou deixar dois códigos para vocês usarem como base.

#### GET:

```
function Get() {  
  return fetch('COLOQUE A URL AQUI!!')  
    .then((response) => response.json())  
    .then((data) => console.log(data));  
}
```

## POST, PUT, DELETE:

```
async function postData(url = '', data = {}) {
  const response = await fetch(url, {
    method: 'POST', // *GET, POST, PUT, DELETE, etc.
    mode: 'cors',
    cache: 'no-cache',
    credentials: 'same-origin',
    headers: {
      'Content-Type': 'application/json'
    },
    redirect: 'follow',
    referrerPolicy: 'no-referrer',
    body: JSON.stringify(data)
  });
  return response.json();
}

postData('COLOQUE URL AQUI', cep /* OBJETO A SER USADO */)
  .then((data) => {
    console.log(data);
  });
```

Em caso de dúvidas recomendo a leitura da seguinte documentação:  
[https://developer.mozilla.org/en-US/docs/Web/API/Fetch\\_API/Using\\_Fetch](https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API/Using_Fetch)