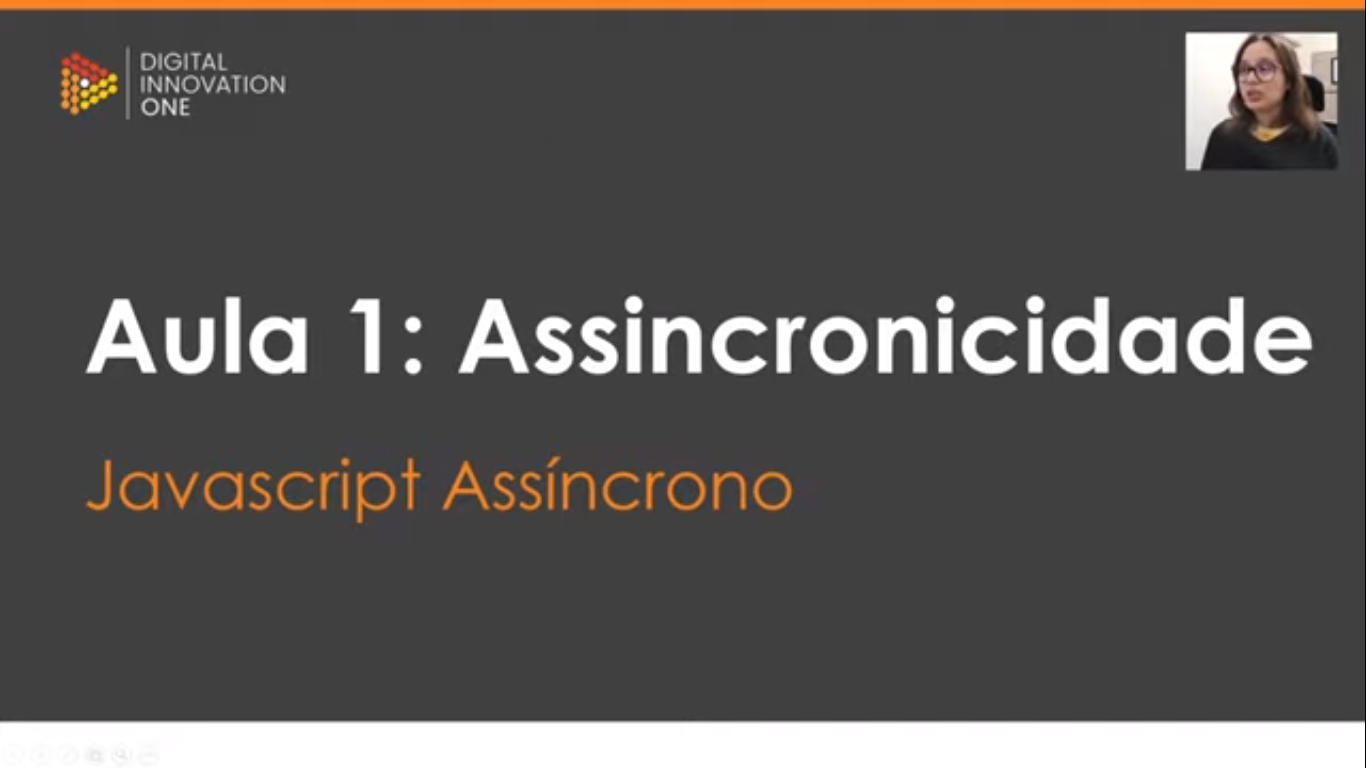
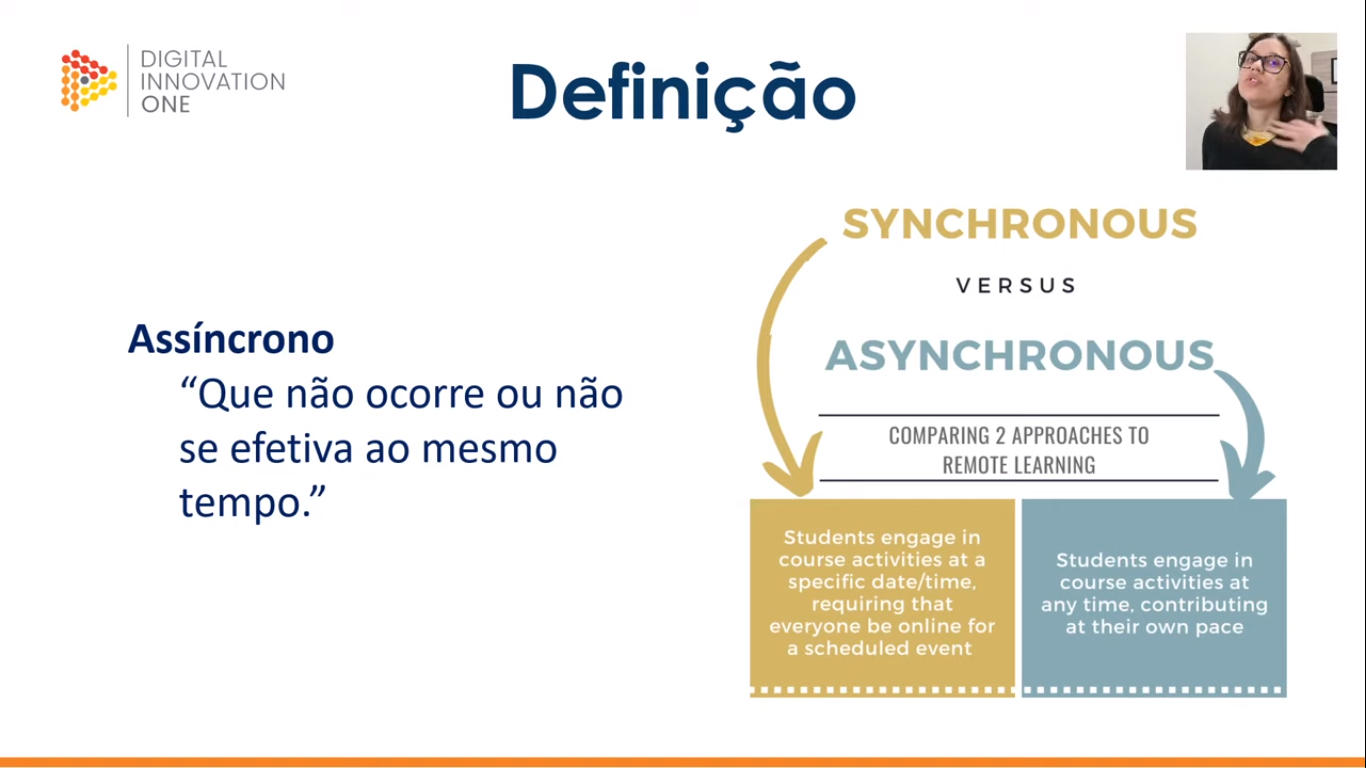
**BOOTCAMP INTER**

**JAVASCRIPT ASSÍNCRONO**





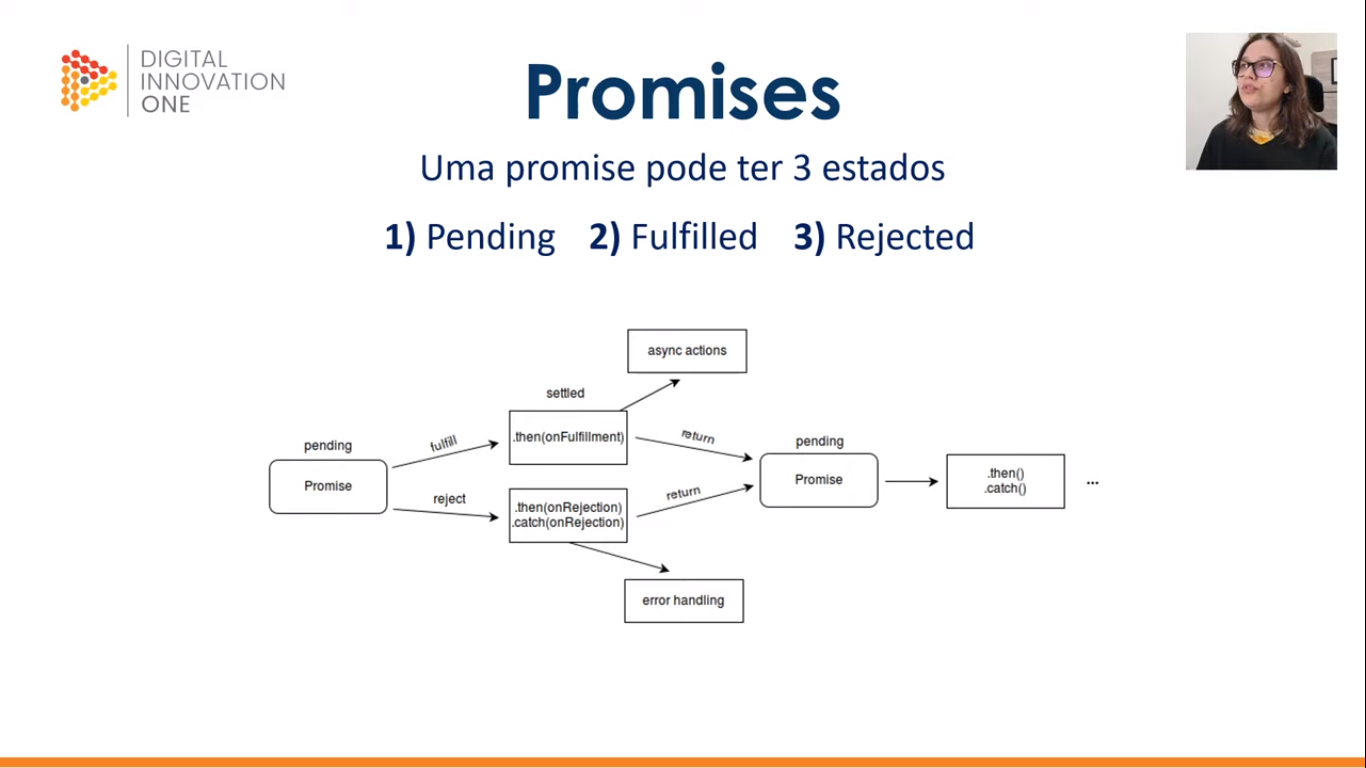








Quando vc compra um ingresso de cinema o que vc tem, em princípio, é apenas uma promessa de que, com aquele ingresso, vc vai assistir o filme. Essa promessa pode ser resolvida, chega na sessão e vc assiste o filme, ou rejeitada, e por qualquer motivo vc não consegue usar o ingresso.



Uma promise quando é declarada está no estado pending e, depois de algum tempo, pode ser rejeitada (rejected) ou completada (fulfilled).



Para declarar uma promise usamos a declaração **new Promise** e especificamos os call-backs para quando for resolvida ou rejeitada.



Para pegar o resultado da Promise depois que ela for resolvida ou rejeitada é preciso utilizar o **await** (que veremos melhor mais a frente).





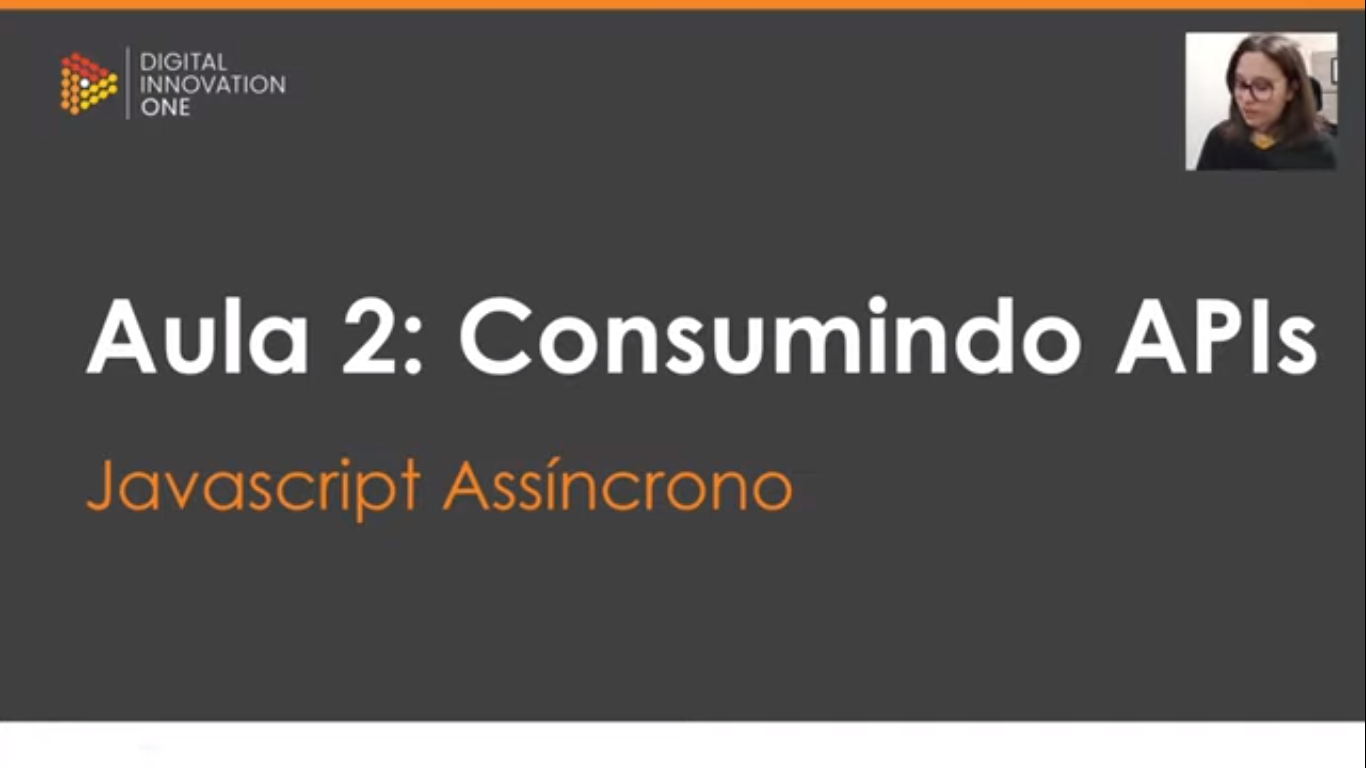
Funções assíncronas devem ser declaradas através da palavra **async** e só se pode usar o **await** para funções assíncronas. O await para o código e espera a promise resolver. Só depois que tiver resolvido é que ele continua o código.



Uma Promise sempre retorna outra Promise. Assim, se eu chamar a resolvePromise sem um await ela vai retornar que está pendente (pending). Para chamar uma função assíncrona eu preciso fazer await resolvePromise.

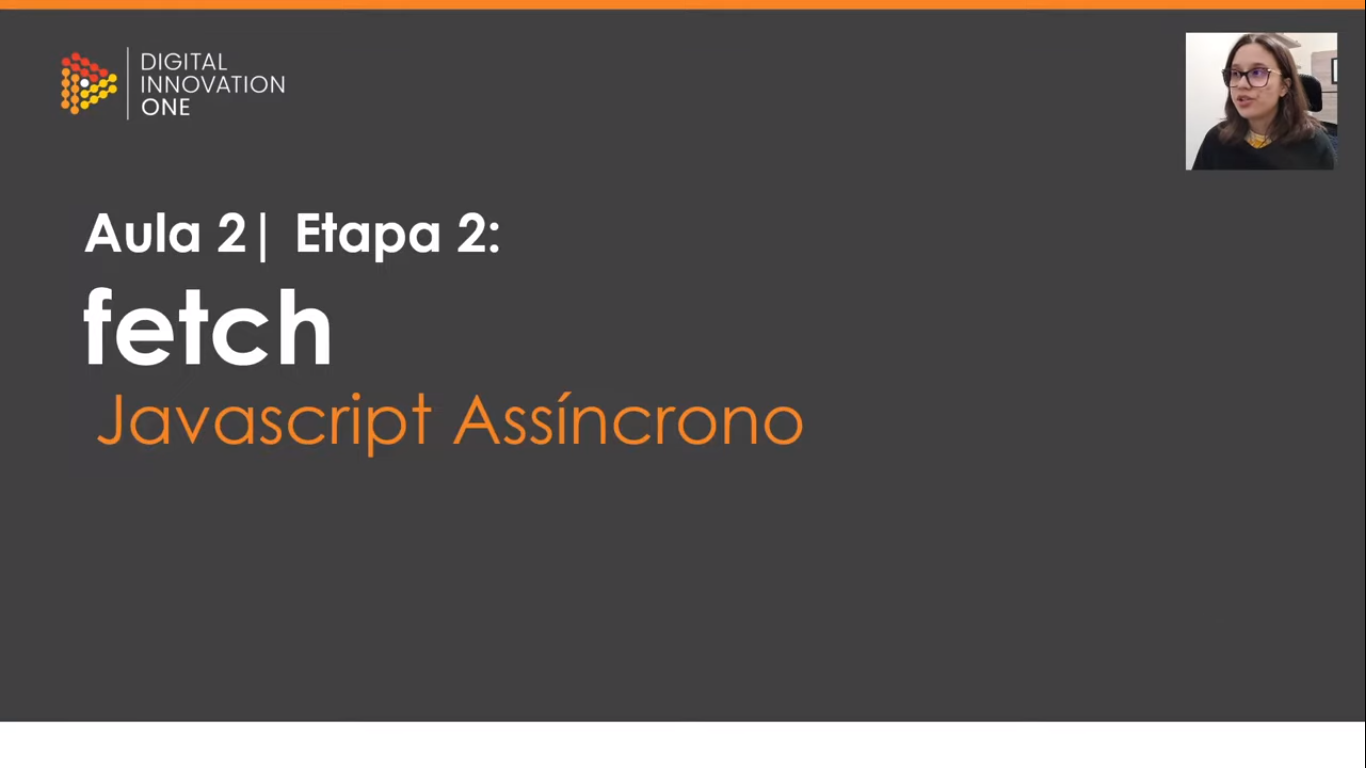


O código anterior também poderia ser escrito utilizando o try...catch, como mostrado acima.



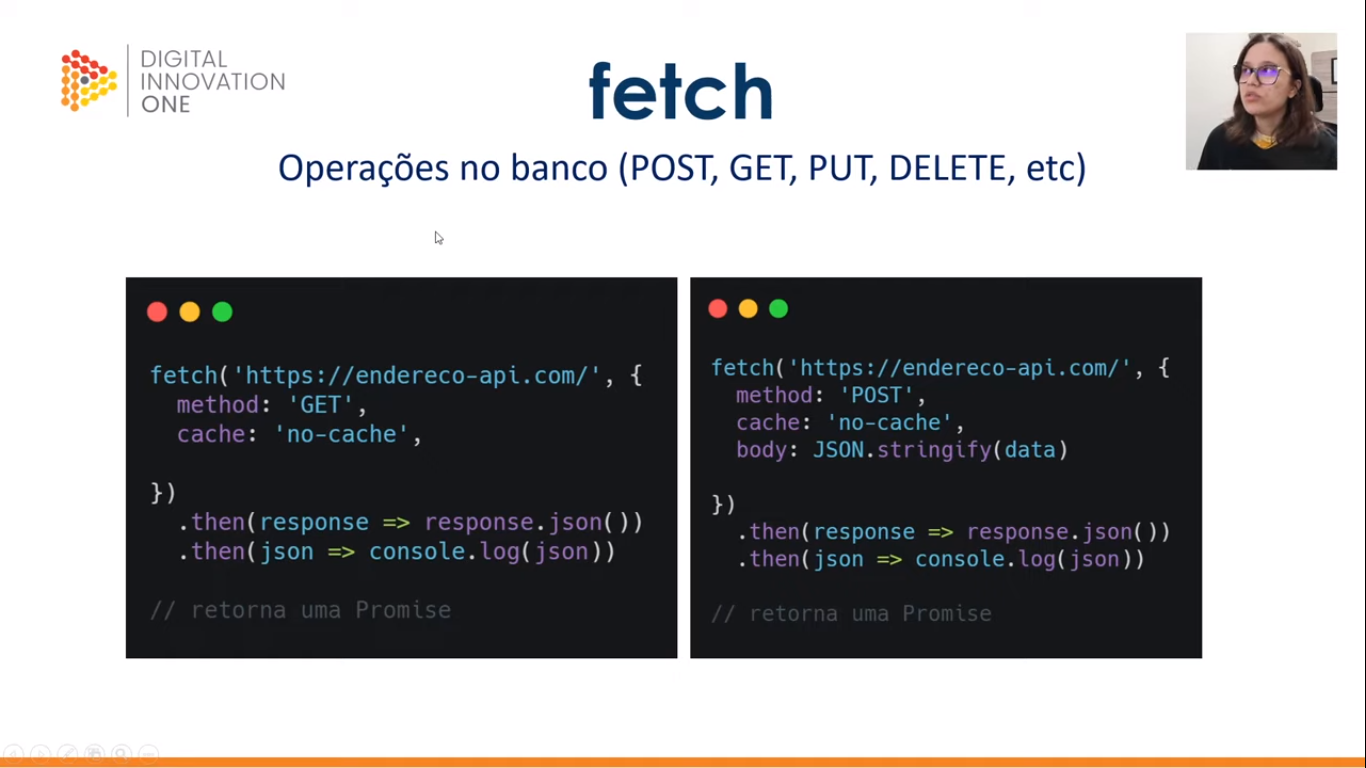




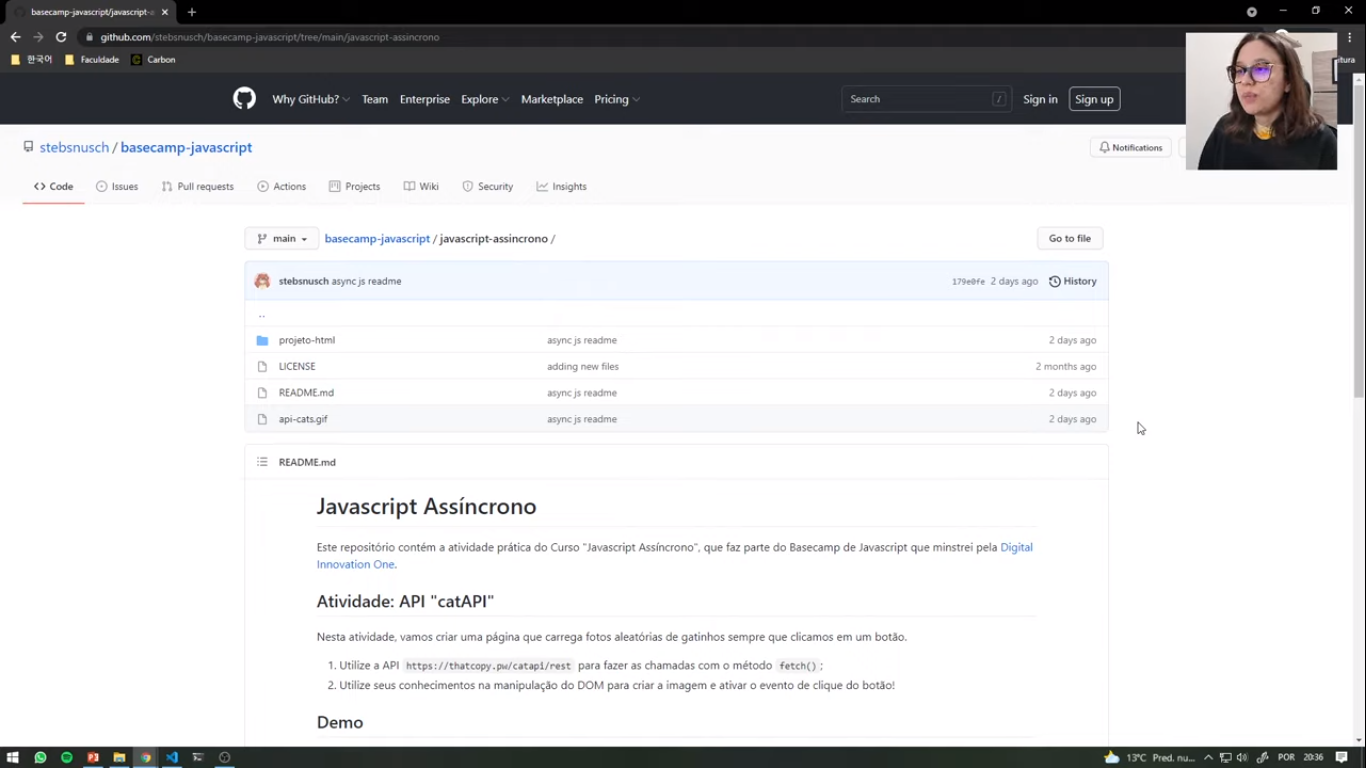


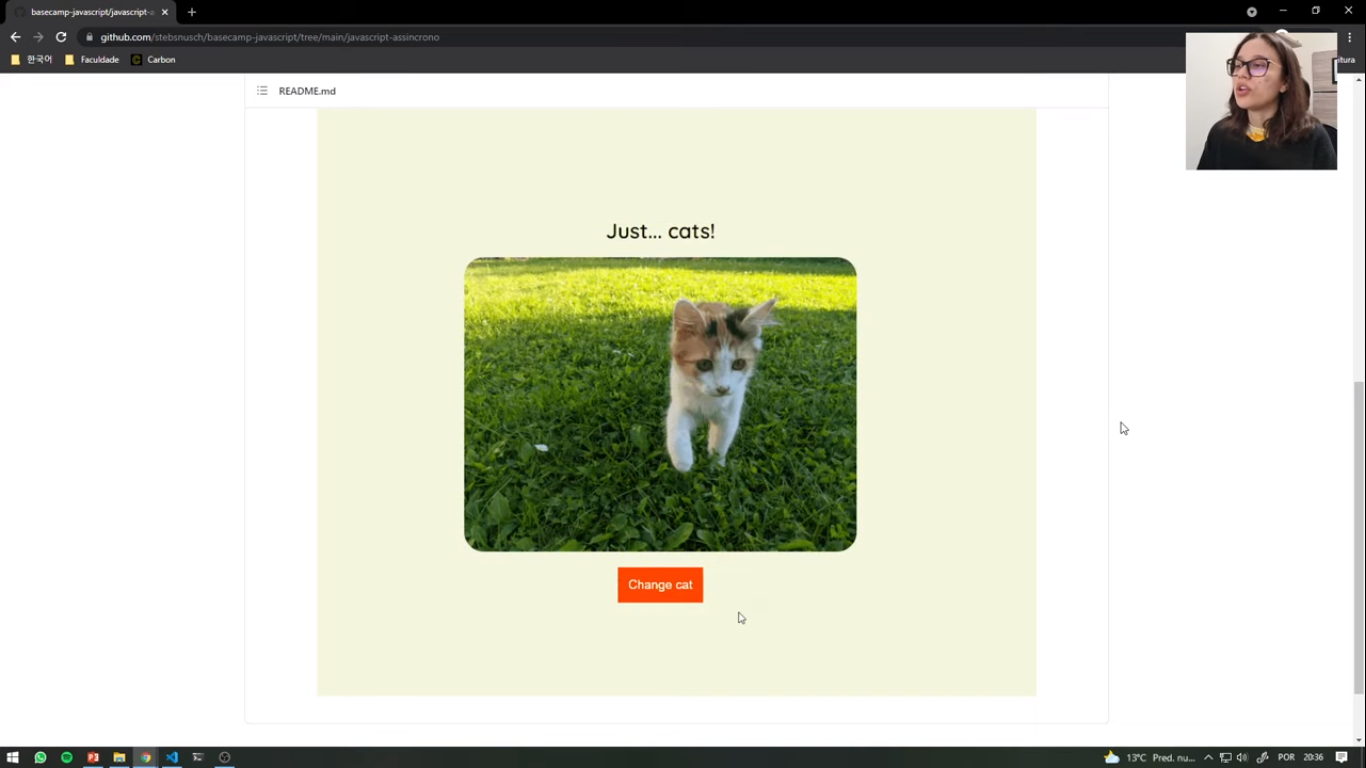


O método **fetch** sempre retorna uma Promise (então lembrar de usar o await quando usar o fetch). É sempre preciso converter a resposta para Json caso ela venha como uma string (texto).



Atividade prática





Clicando no botão ele carrega fotos aleatórias de gatos.

