



# Trio

## Technical Design

This project was given by the Trio Technical Team for me to do within a week. This documentation of the process will help understand my thinking process. It was required to use Expo.

## Background

It is required to create a front end mobile app that displays the menu of a McDonald's franchise. This document exists so I can explain how I created a solution for this problem.

## Technical Approach

I've decided using functional components on this app due to the short time I had for pulling out this project. But I'm also pretty comfortable using it with Hooks. I'm using TypeScript instead of Javascript because of typing. I feel comfortable and I've understood that typing components is a pretty useful way of avoiding some common mistakes that Javascript can have out of the box. I was thinking of using ContextAPI for state management given that Redux was overkill for this project. At the end, used React Hooks.

## Technical Design

### #Handling Promises

```
data: ▼ Promise {_U: 1, _V: 1, _W: {...}, _X: null} ⓘ
  _U: 1
  _V: 1
  ▼ _W:
    currency: "USD"
    description: "Best hamburger of the world! Serving billions with love and quality"
    id: "dc961cfe-dcf2-4ed7-b11b-762b199057fd"
    ▶ location: {lat: '42.356950', log: '-71.059060', address1: 'Washington St', address2: '329', city: 'Boston', ...}
    ▶ menus: (6) [{...}, {...}, {...}, {...}, {...}, {...}]
    name: "McDonald's"
    ▶ operationDays: (7) [{...}, {...}, {...}, {...}, {...}, {...}, {...}]
    phone: "+16175570089"
    privateParking: false
    ▶ [[Prototype]]: Object
  _X: null
  ▶ [[Prototype]]: Object
```

One of my first issues handling this request to the API is that i had a promise. I had to use the async-await solution so I could receive this data and parse it correctly so I could use it.

Solution:

```
fetch(API_URL)
  .then(async res => {
    const data = await res.json();
    if (data) {
      setFranchise(data);
      setLoading(false);
    }
  })
  .catch(() => {
    setLoading(false);
    Alert.alert('Error', 'There was a problem retrieving data');
  });
```

## #Styling

I'm a hardcore flexbox stycler, i don't like using libraries for styling. React Native comes with some cool features for solving styling components. Visually, the most challenging thing about this project was the sliding Modal that shows you the description of the item you want to see once you touch the Card. (I haven't used a `<Card />` component given the short time for doing this project)

## #Constants

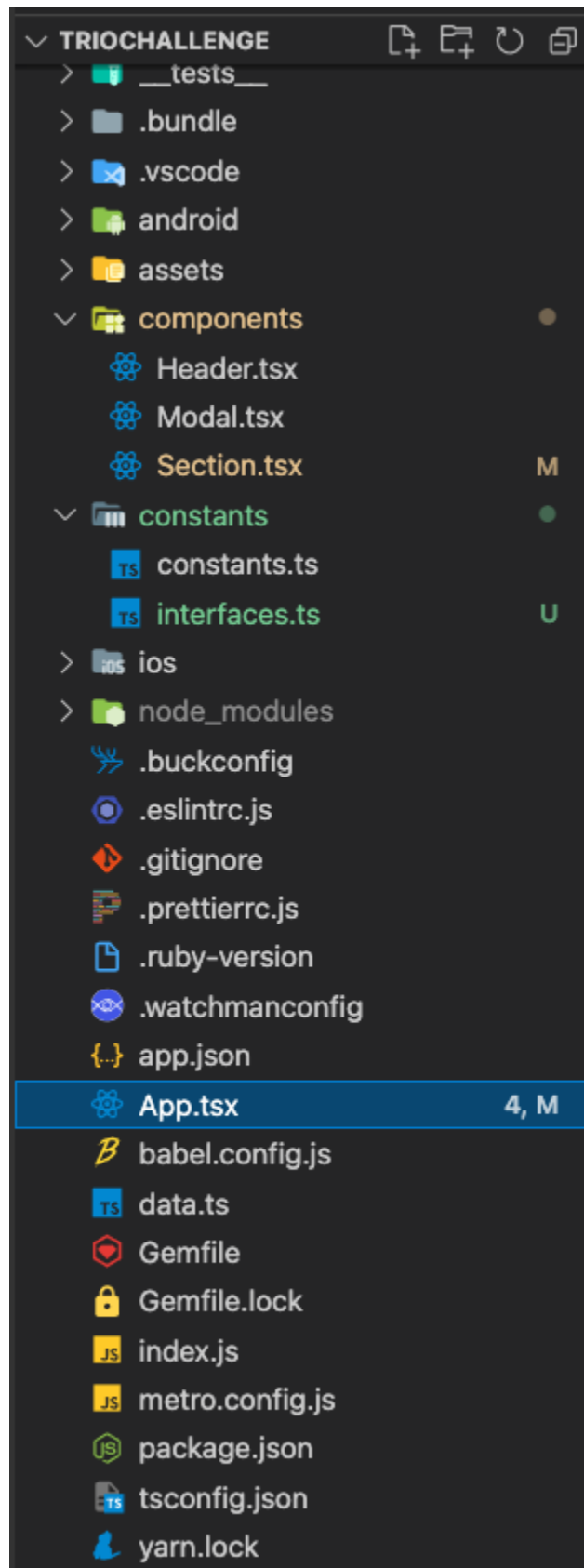
One of my essentials is having a file that can give me a set of constants that can be imported throughout all the project so I don't export Dimensions from react-native all the

time. In this case this was used for handling the image of the item that was displayed on the modal.

## FilePath

This is how I structured the files grouped by components and constants having all of the rendering of the app in the `App.tsx` file.

I started mocking the app with mocked data for the data structure that was provided so I could have the styling of the components created before requesting the data to the API. (see `data.ts` file).

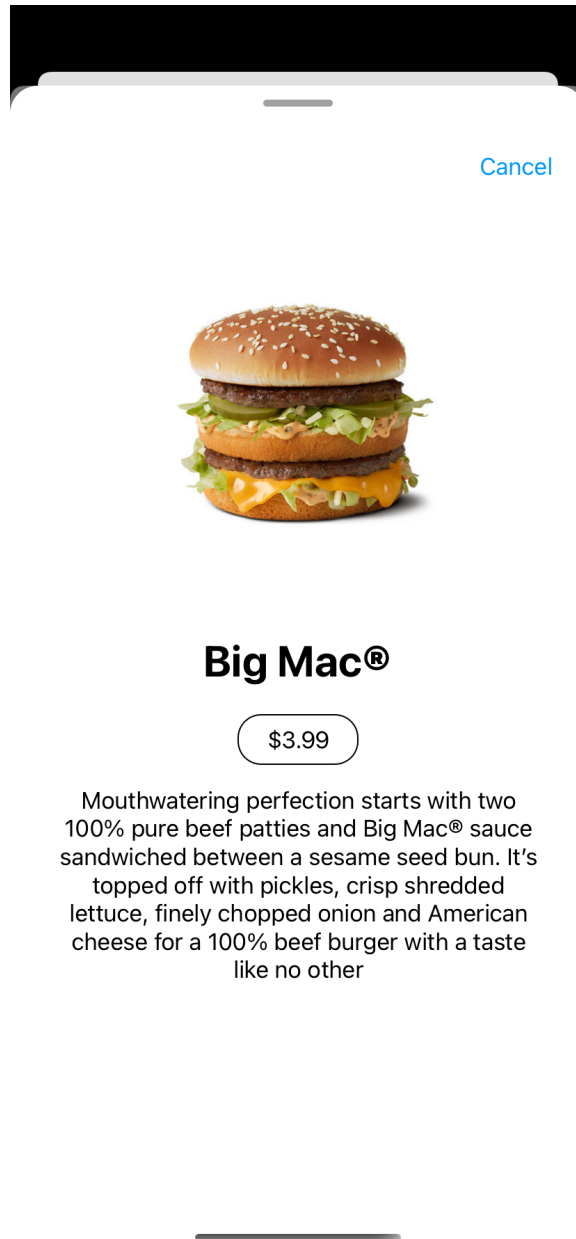


## Difficulties & Solution

I had an issue closing the Modal, specifically with the closing using the dragging down gesture.

The solution that i've found was creating a button similar to the Cancel button on iOS with blueish text color that triggers the function of dismissing the modal.

```
<Pressable
  style={styles.cancel}
  onPress={() => setShowModal(false)}>
  <Text style={styles.txtCancel}>Cancel</Text>
</Pressable>
```



## Questions & Answers

- How does the software work?

React Native is a set of tools that allows me to create hybrid apps for Android and iOS platforms.

- How did you build the project?

I've started creating it based on the designs and the limited timeframe that I had. One of the most important things for me is having the app mocked as soon as

possible. Created mocked data so I can work quickly. I've used Hooks, Flexbox and Typescript as main technologies.

- Why did you build it in that way?

I'm a product based developer. My understanding of how I build mobile apps is product-driven. I try to focus on what the app need to be done the correct way, having in account time (that is money) and what it's needed to be done.

## **Conclusion**

Having a deadline of 7 days, I felt comfortable using React Native for this solution. It was a simple project and I really enjoyed making it. This is my first Technical Design document that I've done and it's like a formal Rubber Duck!