

FeedBack feedApp

O desenvolvimento do app foi desafiador, pois o meu computador não tem suporte suficiente para os requisitos necessários para utilizar o react, enfrentei muitos problemas de configuração de ambiente, e quando finalmente consegui arrumar, o computador travava o tempo todo devido o visual studio ser muito grande, e para rodar o react no windows, é necessário ter um simulador rodando, tentei também usar a ferramenta disponível online para desenvolvimento de react, porém foi bem sucedida. Depois de um dia de tentativa, resolvi mudar de framework, escolhi o ionic 3, por ter uma familiaridade com ele e por ser bem mais simples de simular, pois roda até mesmo no navegador.

Com a mudança de framework, consegui iniciar o desenvolvimento do app, porém enfrentei um novo desafio, apesar do ionic proporcionar testes no navegador, para testar funções nativas (como uso de câmera) é necessário o simulador, a qual meu computador não tem suporte para processar, implementei o upload de fotos, porém não foi possível efetuar o teste, abaixo descrevo o que desenvolvido

desenvolvimento do feedApp

1. Preparação do ambiente de desenvolvimento.(necessário para rodar aplicação
 - a. Instalação da ionic 3, necessário o cordova
 - b. instalação do Node.js, knexjs, nodemon, node-restify
2. O que foi desenvolvido no Front-End
 - a. Menu superior: em abas, seguindo a imagem
 - b. Página de Feeds: listar todos, adicionar, detalhar
 - c. Página de Avisos: Listar avisos, adicionar
 - d. Página Parcerias: listar benefícios oferecidos, adicionar
 - e. Página de Login: não estava no protótipo, porém resolvi inserir, pois dessa forma a restrição para suprir as regras de negócio são facilitadas.
3. O que foi desenvolvido no Back-End
 - a. Banco de dados
 - i. diagrama no *workbench*
 - ii. Organização do banco: devido às regras de negócios estabelecida para o aplicativo, com restrição de somente alguns usuários fazer adição, resolvi centralizar a tabela de usuário, tendo um perfil, dessa forma, é possível controlar permissões a partir somente de uma tabela.
 - b. Api's - A informação de acesso às apis está no fim deste documento
 - i. Node.js utilizando o restify e banco de dados mysql.
 - ii. Api's para o feed: adicionar, excluir, listar, editar
 - iii. Api's para aviso: adicionar, excluir, listar, editar
 - iv. Api's para parcerias: adicionar, excluir, listar e editar
 - v. Api's de usuário: adicionar, excluir, listar e editar
 - c. Integração das api's com o aplicativo

- i. Integração foi feita utilizando HttpClient do próprio Ionic 3.
 - ii. Neste passo a dificuldade enfrentada foi a de Cors, porém, foi superada.
 - iii. Para visualizar como foi feito essa integração, o arquivo: home.ts, obtém algumas chamadas
- d. Regras de negócio
 - i. *Somente setor do endomarketing pode postar nos avisos*: controle este está sendo feito através do perfil, usei adm como sendo endomarketing.
 - ii. *Validar entrada/saída*: Validação de formulários sendo feita através da desativação do botão enquanto não for adicionado todos campos obrigatórios
 - iii. *Colaboradores somente visualizam os avisos e parcerias*: controle está sendo feito através do perfil do usuário, usei func, como sendo os funcionários

Chamadas para APIS

1. Adicionar feeds/avisos/parcerias/usuarios

a. url: [http://localhost:8080/\[nomeTabela\]/create](http://localhost:8080/[nomeTabela]/create)

b. entrada:

```
{
  "titulo": "Lotado",
  "url_imagem": "",
  "legenda": "venham, aqui ta cheinho de gentee",
  "usuarios_id": "1"
}
```

c. saída

```
[
  3
]
```

2. Listar feeds/avisos/parcerias/usuarios

a. url: [http://localhost:8080/\[nomeTabela\]](http://localhost:8080/[nomeTabela])

b. entrada: nenhuma

c. saída:

```
[
  {
```

```

      "id": 2,
      "titulo": "Promoção de fim de ano bombando",
      "url_imagem": "",
      "legenda": "venham, aqui ta cheinho de gentee",
      "usuarios_id": 1
    },
    {
      "id": 3,
      "titulo": "Lotado",
      "url_imagem": "",
      "legenda": "venham, aqui ta cheinho de gentee",
      "usuarios_id": 1
    }
  ]

```

3. View feed/aviso/parceria/usuario

- a. link: [http://localhost:8080/\[nomeTabela\]/view/\[id\]](http://localhost:8080/[nomeTabela]/view/[id])
- b. entrada: nenhum
- c. saída: {


```

        "id": 2,
        "titulo": "Promoção de fim de ano bombando",
        "url_imagem": "",
        "legenda": "venham, aqui ta cheinho de gentee",
        "usuarios_id": 1
      
```

4. Editar feeds/avisos/parcerias/usuarios

- a. link: [http://localhost:8080/\[nomeTabela\]/update/\[id\]](http://localhost:8080/[nomeTabela]/update/[id])
- b. entrada: mesmos campos do adicionar
- c. saída: sucesso ou erro

5. Login do Usuário

- a. link: <http://localhost:8080/usuario/login/:nome/:senha>
- b. entrada: nenhuma
- c. saída: nome, perfil e senha