



Mesa  
Livre

**Aplicativo para gerenciar filas e mesas**

Gustavo de Freitas, Lucas Borba, Tiago Salles, Vinicius

# Ideia principal

**Melhorar os atendimentos recorrentes na fila de espera de restaurantes.**

O aplicativo foi desenvolvido para facilitar a gestão de filas, evitando acúmulos causados pela falta de organização.

# Funcionalidades

## Organização das mesas:

Esta aplicação concede aos donos de restaurantes a possibilidade de estruturar e administrar os espaços de suas mesas, além de possibilitar a alocação de mesas para os clientes da fila.

## Gerenciamento de filas:

O aplicativo oferece aos gestores de restaurantes a possibilidade de gerir as listas de espera. Isso inclui a capacidade de adicionar ou remover clientes da fila, bem como atribuir prioridades de atendimento para clientes específicos.

# Nome e marca do App



Mesa  
Livre

# Aplicativo Mobile

## React native

A biblioteca mais amplamente utilizada no mercado de trabalho para o desenvolvimento de interfaces de usuário é o React.js.

Com o React.js, é possível utilizar o framework React Native para desenvolver aplicativos móveis.

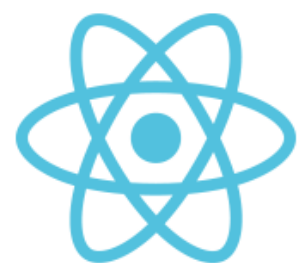
Este framework foi escolhido pelas suas principais funcionalidades:

- Desenvolvimento Multiplataforma
- Desempenho Próximo ao Nativo
- Comunidade Ativa e Suporte Extenso:

# Dependências



A X 1 O S



Vector Icons



Choose from 3K+ icons or use your own

## Componentes

react-native-raw-bottom-sheet [TS](#)

3.0.0 • [Public](#) • Published 3 months ago

[Readme](#)

[Code](#) [Beta](#)

react-native-raw-bottom-sheet

react-native-paper-select [TS](#)

1.1.1 • [Public](#) • Published a month ago

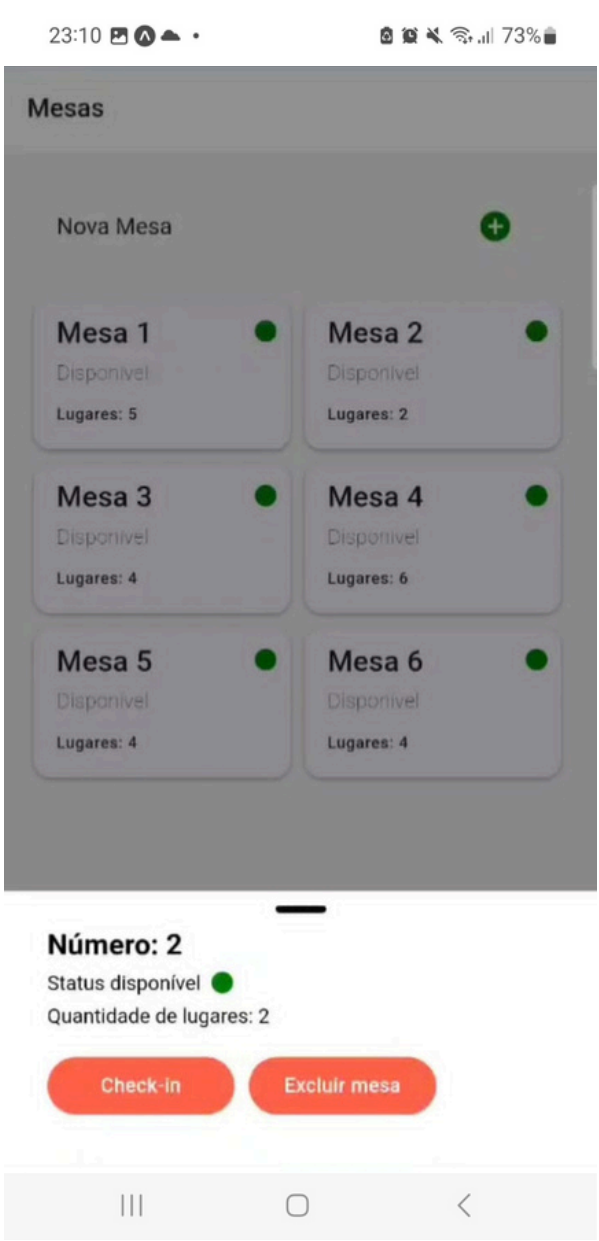
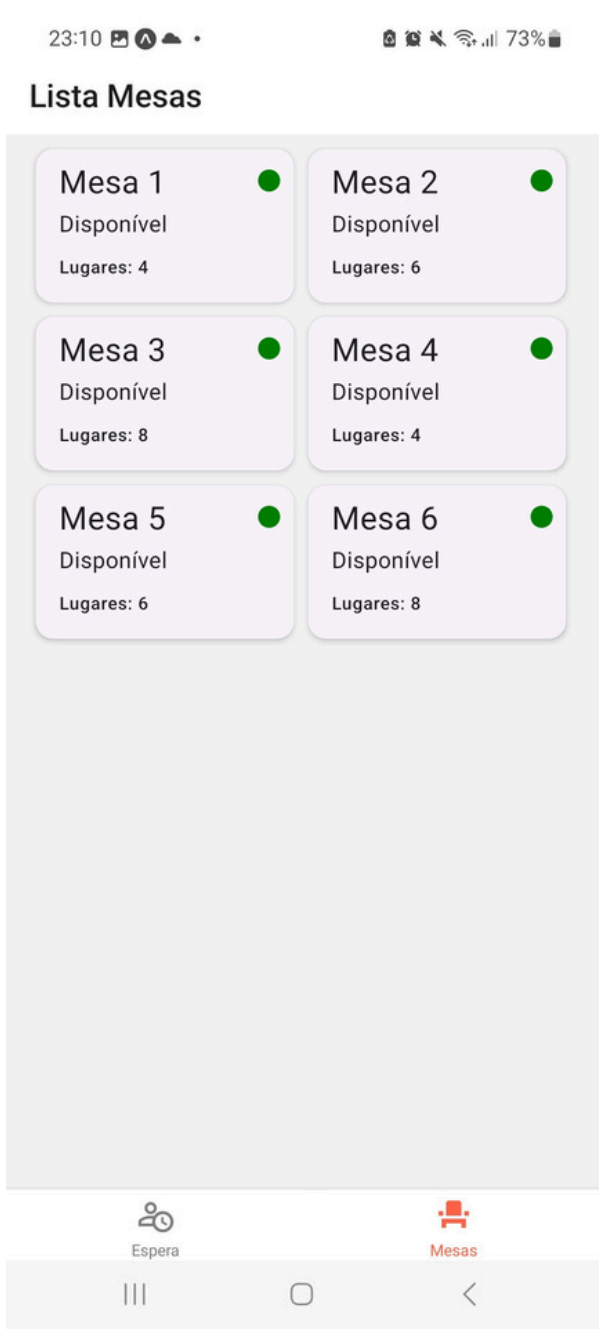
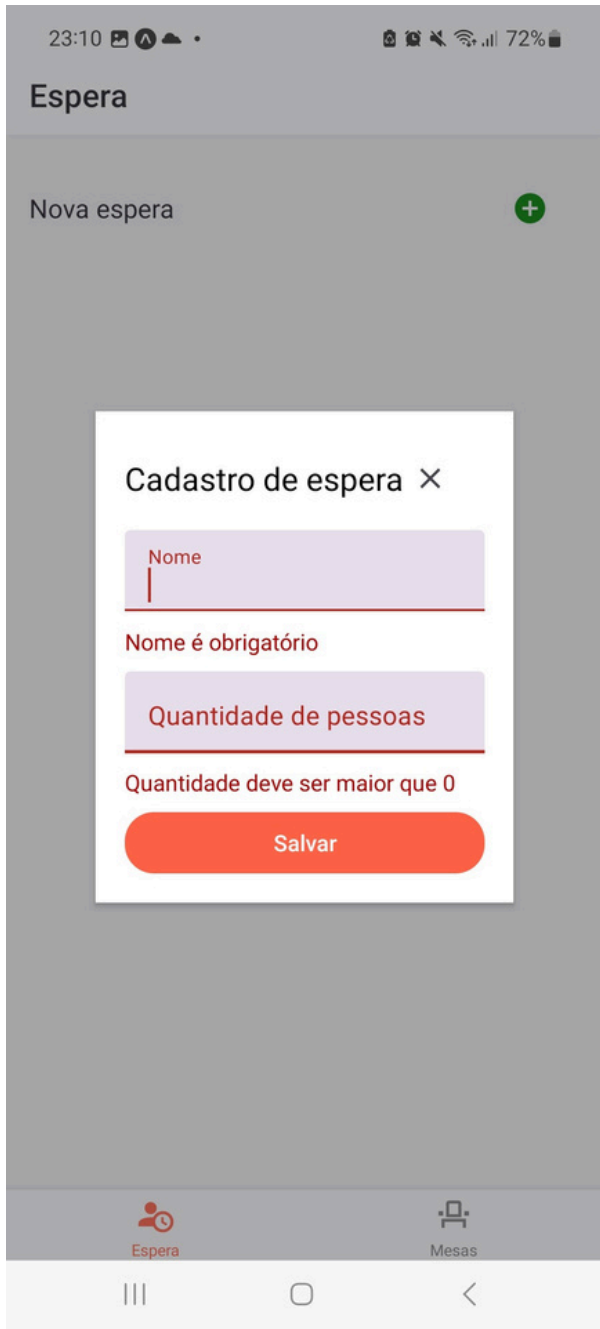
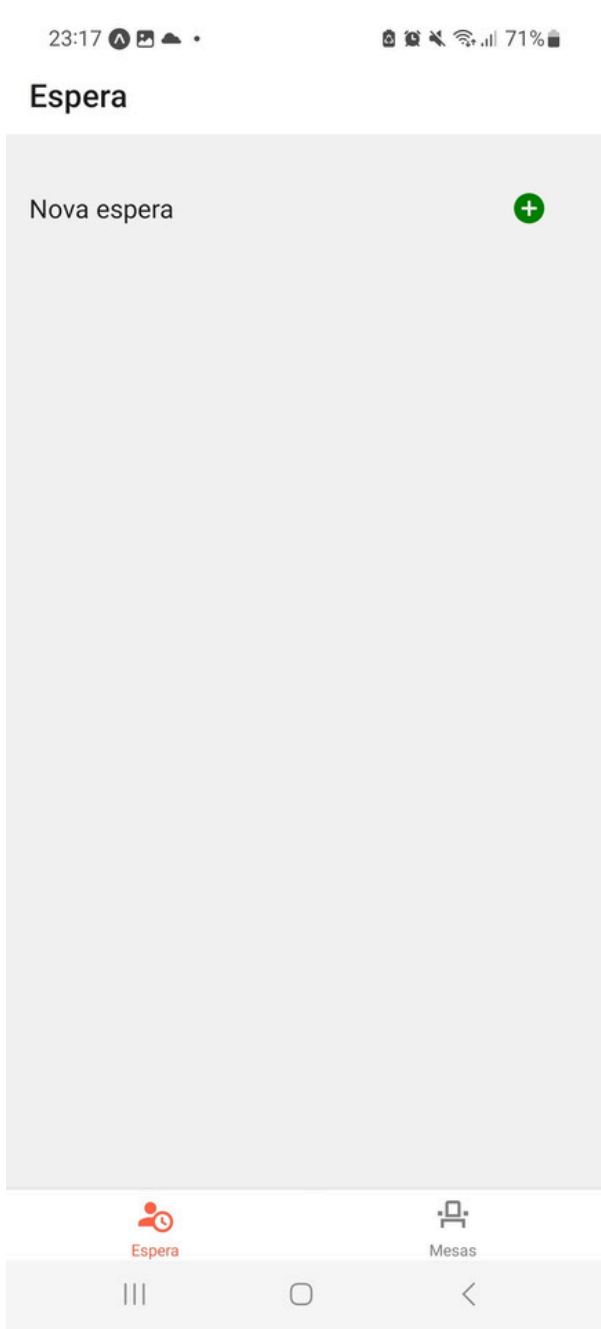
[Readme](#)

[Code](#) [Beta](#)

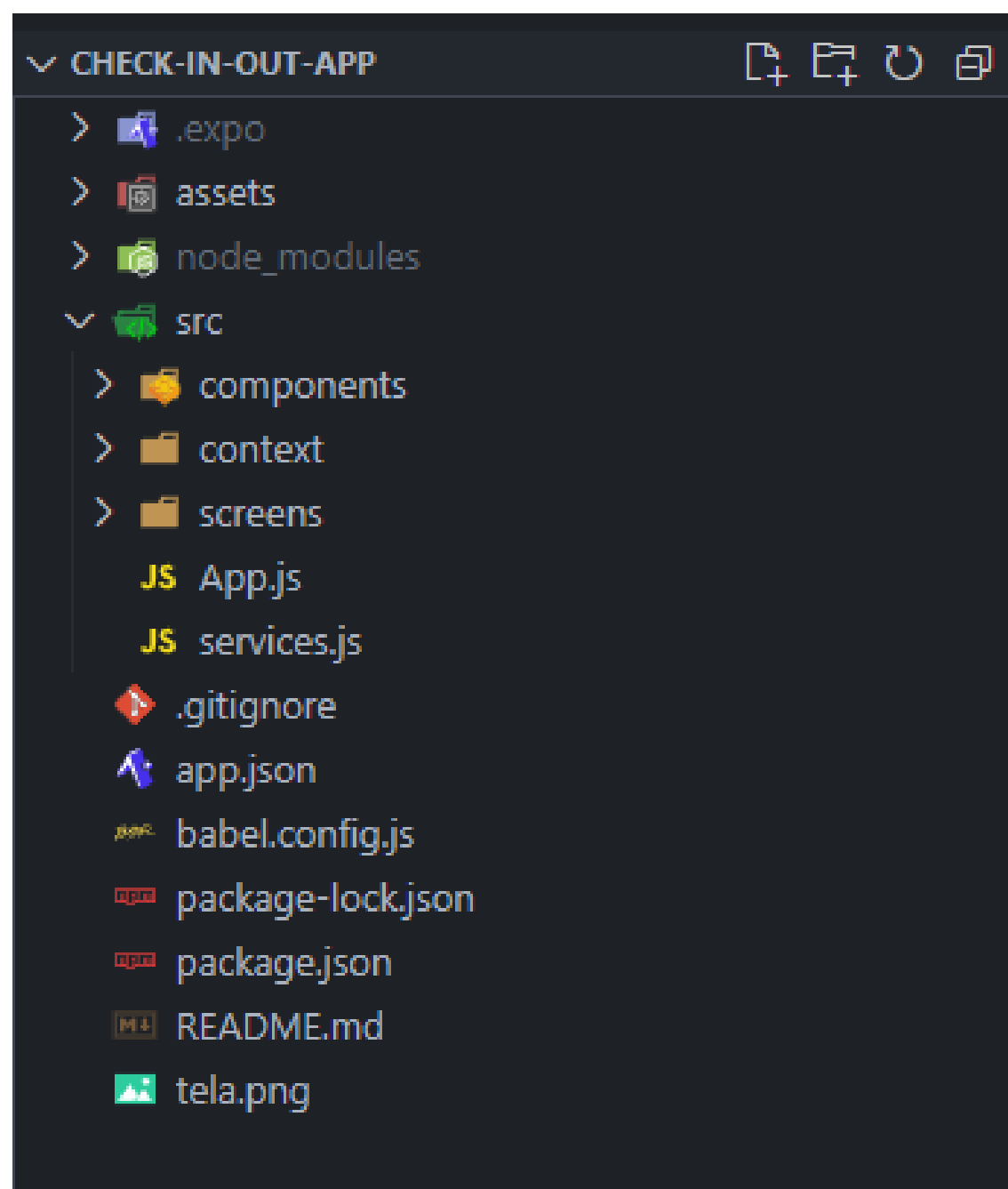
React Native Paper Select [▼](#)



# Telas



# Árvore do projeto





# Persistência dos dados

## API REST

```
1 <?php
2
3 header("Access-Control-Allow-Origin: *");
4 header("Content-Type: application/json; charset=UTF-8");
5
6 require 'database.php';
7
8 if (isset($_POST['nome_cliente'], $_POST['qtd_pessoas'], $_POST['id_mesa']))
9 {
10     $nome_cliente = $_POST['nome_cliente'];
11     $qtd_pessoas = $_POST['qtd_pessoas'];
12     $id_mesa = $_POST['id_mesa'];
13
14     try {
15         $stmt = $conn->prepare("INSERT INTO clientes (nome_cliente, qtd_pessoas, id_mesa,
16         $stmt->execute();
17         $stmt1 = $conn->prepare("UPDATE mesas SET reserva = 1 WHERE id_mesa = $id_mesa");
18         $stmt1->execute();
19         $clienteId = $conn->lastInsertId();
20         echo json_encode(['id_cliente' => $clienteId, 'nome_cliente' => $nome_cliente, 'q
21     } catch(PDOException $e) {
22         echo json_encode(['error' => $e->getMessage()]);
23     }
24 }
25
26
```



## Database

```
4
3
4 • SELECT * FROM clientes
5
```

	id_cliente	nome_cliente	qtd_pessoas	id_mesa	data_ini	data_fim	espera
▶	1	teste	4	6	2024-05-01 18:00:38	NULL	0
	2	joaozinho	2	1	2024-05-18 00:00:00	NULL	0
	5	pedrinho	2	3	2024-05-18 22:16:52	NULL	0
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL



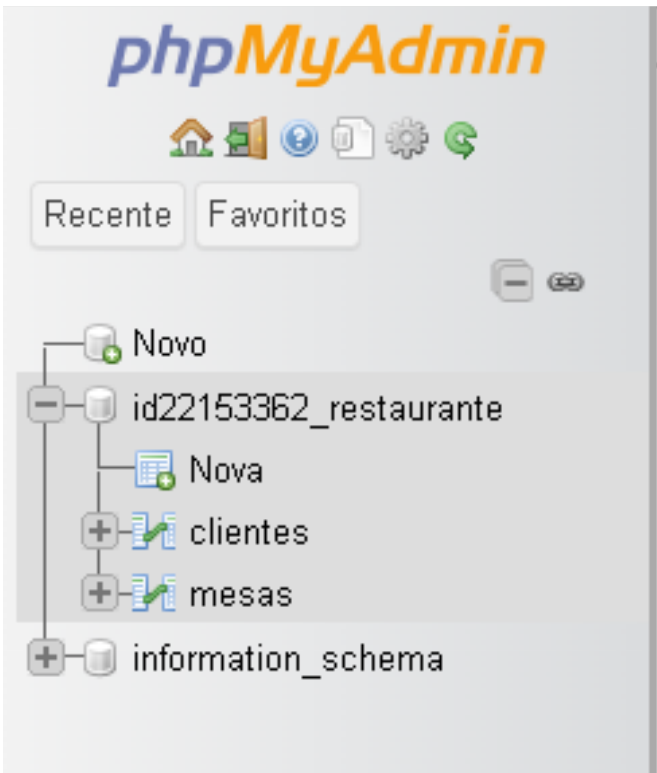
# Hospedagem (API / DATABASE)

- 000webhost  000webhost  
powered by HOSTINGER

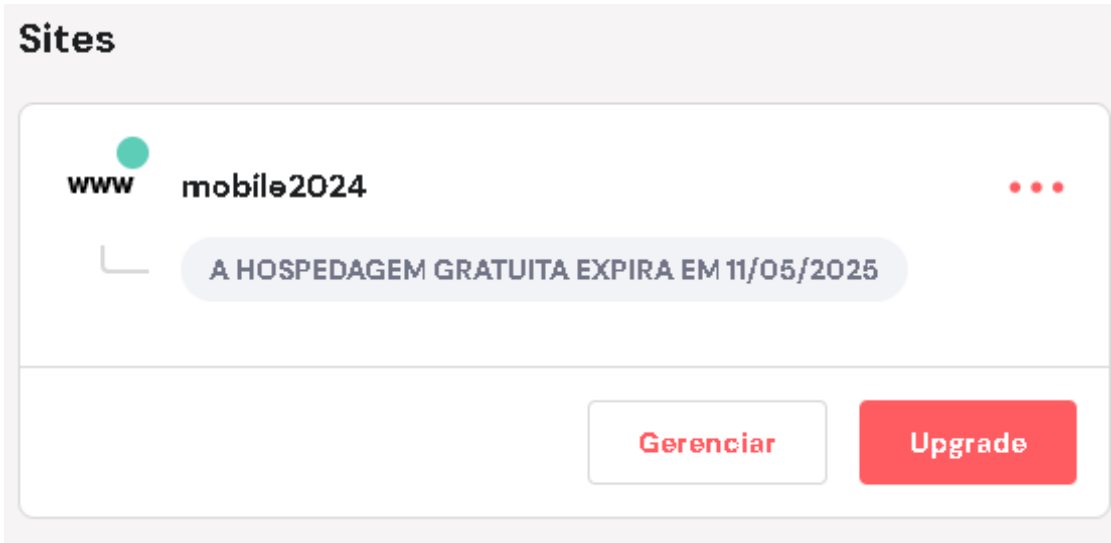
## Endpoints utilizados

<input type="checkbox"/>	Name ▼
<input type="checkbox"/>	.htaccess
<input type="checkbox"/>	checkin_cliente_da_fila.php
<input type="checkbox"/>	clientes_por_qtd.php
<input type="checkbox"/>	database.php
<input type="checkbox"/>	enviar_cliente_fila.php
<input type="checkbox"/>	excluir_cliente_fila.php
<input type="checkbox"/>	excluir_cliente_mesa.php
<input type="checkbox"/>	fila_de_espera.php
<input type="checkbox"/>	listar_mesas.php

## Gerenciador - database



## Site dos endpoints




<https://mobile2024.000webhostapp.com/>

# Consumindo API

INSERT



 checkin\_cliente.php

Cadastro de espera ✕

0|


Salvar

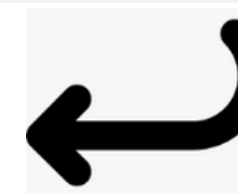
← Mesa 1

Status reservada ●

Quantidade de lugares: 4

Check-out

 excluir\_cliente\_mesa.php



DELETE

# Demonstração do aplicativo