

UC: Projetos

Assunto:

Projeto Pizzaria – Frontend_02 (NextJS)

Sistema Pizzaria Web

57 – Exibindo Pedidos

frontend

EXPLORER

FRONTEND

- > .next
- > node_modules
- > public
- src
 - app
 - dashboard
 - > category
 - > components
 - > product
 - layout.tsx
 - page.tsx
 - > fonts
 - > signup
 - ★ favicon.ico
 - ♀ globals.scss
 - layout.tsx
 - ♀ page.module.scss
 - page.tsx
- lib
 - TS cookieClient.ts
 - TS cookieServer.ts
 - TS order.type.ts
- > services
- TS middleware.ts
- diamond .gitignore
- TS next-env.d.ts

OUTLINE

TIMELINE

page.tsx

src > app > dashboard > page.tsx > Dashboard

```
9  const response = await api.get("/orders", {
10    headers: {
11      Authorization: `Bearer ${token}`
12    }
13  })
14  return response.data || [];
15}
16} catch(err){
17  console.log(err);
18  return [];
19}
20}
21}
22}

export default async function Dashboard(){
23
24  const orders = await getOrders();
25
26  console.log(orders);
27
28  return(
29    <>
30    <Orders/>
31    </>
32  )
33}
34}
35}
```

Ln 28, Col 3 (20 selected) Spaces: 2 UTF-8 CRLF (TypeScript JSX ✓ Prettier

Sujeito Pizza - A melhor pizzaria

localhost:300...

SujeitoPizza

Categoria Produto [→]

Últimos pedidos

- Mesa 10
- Mesa 13

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure for a Next.js application named "frontend".
- File:** `page.tsx` is the active file.
- Content:** The code is an asynchronous function `Dashboard` that fetches orders from an API using `api.get("/orders")`. It handles errors and returns an empty array if no orders are found. It then renders a component `<Orders>`.

```
const response = await api.get("/orders", {
  headers: {
    Authorization: `Bearer ${token}`
  }
});

return response.data || [];

} catch(err) {
  console.log(err);
  return [];
}

export default async function Dashboard() {
  const orders = await getOrders();

  return(
    <>
      <Orders/>
    </>
  )
}
```

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localhost:300...
- Content:** Displays the title "SujeitoPizza" and a section titled "Últimos pedidos" (Last orders) showing "Mesa 10" and "Mesa 13".

frontend

page.tsx index.tsx

src > app > dashboard > components > orders > index.tsx > Orders

```
4  export function Orders(){
5    return(
6      <main className={styles.container}>
7
8        <section className={styles.containerHeader}>
9          <h1>Últimos pedidos</h1>
10         <button>
11           <RefreshCw size={24} color="#3ffffa3" />
12         </button>
13       </section>
14
15       <section className={styles.listOrders}>
16
17         <button
18           className={styles.orderItem}>
19             <div className={styles.tag}></div>
20             <span>Mesa 10</span>
21           </button>
22
23         <button
24           className={styles.orderItem}>
25             <div className={styles.tag}></div>
26             <span>Mesa 13</span>
27           </button>
28
29       </section>
30
31     </main>
32
33   }
```

Ln 27, Col 45 Spaces: 2 UTF-8 CRLF TypeScript JSX Prettier

Sujeito Pizza - A melhor pizzaria

localhost:300...

SujeitoPizza

Categoria Produto

Últimos pedidos

Mesa 10

Mesa 13

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "page.tsx" is the active file.
- Content:** The code is a TypeScript component named "Orders". It imports styles from "styles.module.scss", a refresh icon from "lucide-react", and an interface "OrderProps" from "@lib/order.type". The component has a prop "orders" of type "OrderProps[]". It contains two sections: "Últimos pedidos" and "listOrders". Each "listOrders" item is a button with a "tag" class containing the text "Mesa 10".

```
src > app > dashboard > components > orders > index.tsx > Orders
1 import styles from './styles.module.scss'
2 import { RefreshCw } from 'lucide-react'
3 import { OrderProps } from '@/lib/order.type'
4
5 interface Props{
6   orders: OrderProps[]
7 }
8
9 export function Orders({ orders }: Props){
10   return(
11     <main className={styles.container}>
12       <section className={styles.containerHeader}>
13         <h1>Últimos pedidos</h1>
14         <button>
15           <RefreshCw size={24} color="#3ffffa3" />
16         </button>
17       </section>
18
19       <section className={styles.listOrders}>
20
21         <button
22           className={styles.orderItem}
23         >
24           <div className={styles.tag}></div>
25           <span>Mesa 10</span>
26         </button>
27
28         <button
29           className={styles.orderItem}
30         >
31           <div className={styles.tag}></div>
```

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localhost:300...
- Content:** The page displays the heading "SujeitoPizza" and a section titled "Últimos pedidos" with two items: "Mesa 10" and "Mesa 13".

frontend

EXPLORER

FRONTEND

- > .next
- > node_modules
- > public
- src
 - app
 - dashboard
 - category
 - components
 - button
 - header
 - orders
 - index.tsx
 - styles.module.scss
 - product
 - layout.tsx
 - page.tsx 1
 - fonts
 - signup
 - favicon.ico
 - globals.scss
 - layout.tsx
 - page.module.scss
 - page.tsx
- lib
 - TS cookieClient.ts
 - TS cookieServer.ts

OUTLINE

TIMELINE

page.tsx 1 X index.tsx

src > app > dashboard > page.tsx > Dashboard

```
9      const response = await api.get("/orders", {  
10     headers: {  
11       Authorization: `Bearer ${token}`  
12     }  
13   })  
14   return response.data || [];  
15 }catch(err){  
16   console.log(err);  
17   return [];  
18 }  
19  
20 export default async function Dashboard(){  
21   const orders = await getOrders();  
22  
23   return(  
24     <>  
25     <Orders>  
26     </>  
27   )  
28 }  
29  
30 }  
31  
32 }  
33 }
```

Ln 30, Col 6 (6 selected) Spaces: 2 CRLF TypeScript JSX Prettier

Sujeito Pizza - A melhor pizzaria

localhost:300...

SujeitoPizza

Categoria Produto

Últimos pedidos

- Mesa 10
- Mesa 13

frontend

EXPLORER

FRONTEND

- > .next
- > node_modules
- > public
- src
 - app
 - dashboard
 - > category
 - > components
 - > button
 - > header
 - > orders
 - index.tsx
 - styles.module.scss
 - > product
 - layout.tsx
 - page.tsx
 - > fonts
 - > signup
 - ★ favicon.ico
 - globals.scss
 - layout.tsx
 - page.module.scss
 - page.tsx

lib

 - TS cookieClient.ts
 - TS cookieServer.ts

OUTLINE

TIMELINE

page.tsx X index.tsx

src > app > dashboard > page.tsx > Dashboard

```
9  const response = await api.get("/orders", {
10    headers: {
11      Authorization: `Bearer ${token}`
12    }
13  })
14  return response.data || [];
15}
16} catch(err){
17  console.log(err);
18  return [];
19}
20}
21}
22}
23}
24export default async function Dashboard(){
25  const orders = await getOrders();
26
27  return(
28    <>
29    <Orders orders={orders}/>
30    </>
31  )
32}
33}
```

Ln 30, Col 28 (15 selected) Spaces: 2 CRLF TypeScript JSX Prettier

Sujeito Pizza - A melhor pizzaria

localhost:300...

SujeitoPizza

Categoria Produto

Últimos pedidos

- Mesa 10
- Mesa 13

frontend

EXPLORER

FRONTEND

- .next
- node_modules
- public
- src
 - app
 - dashboard
 - category
 - components
 - button
 - header
 - orders
 - index.tsx
 - styles.module.scss
 - product
 - layout.tsx
 - page.tsx
 - fonts
 - signup
 - favicon.ico
 - globals.scss
 - layout.tsx
 - page.module.scss
 - page.tsx
 - lib
 - cookieClient.ts
 - cookieServer.ts

OUTLINE

TIMELINE

page.tsx index.tsx

src > app > dashboard > components > orders > index.tsx > Orders

```
orders: OrderProps[]

export function Orders({ orders }: Props) {
  return(
    <main className={styles.container}>
      <section className={styles.containerHeader}>
        <h1>Últimos pedidos</h1>
        <button>
          <RefreshCw size={24} color="#3fffa3" />
        </button>
      </section>

      <section className={styles.listOrders}>
        <button
          className={styles.orderItem}
        >
          <div className={styles.tag}></div>
          <span>Mesa 10</span>
        </button>

        <button
          className={styles.orderItem}
        >
          <div className={styles.tag}></div>
          <span>Mesa 13</span>
        </button>
      </section>
    </main>
  )
}
```

Ln 21, Col 1 (335 selected) Spaces: 2 CRLF TypeScript JSX ✓ Prettier

Sujeito Pizza - A melhor pizzaria

localhost:300...

SujeitoPizza

Categoria Produto [→]

Últimos pedidos

- Mesa 10
- Mesa 13

```
page.tsx index.tsx
src > app > dashboard > components > orders > index.tsx > Orders
orders: OrderProps[]

export function Orders({ orders }: Props) {
  return(
    <main className={styles.container}>
      <section className={styles.containerHeader}>
        <h1>Últimos pedidos</h1>
        <button>
          <RefreshCw size={24} color="#3fffa3" />
        </button>
      </section>
      <section className={styles.listOrders}>
        </section>
    </main>
  )
}

Ln 21, Col 1  Spaces: 2  UTF-8  CRLF  TypeScript JSX  Prettier
```

SujeitoPizza

Categoria Produto

Últimos pedidos

↻

frontend

EXPLORER

FRONTEND

- > .next
- > node_modules
- > public
- src
 - app
 - dashboard
 - > category
 - > components
 - > button
 - > header
 - > orders
 - index.tsx
 - styles.module.scss

index.tsx

```
src > app > dashboard > components > orders > index.tsx > Orders
orders: OrderProps[]

export function Orders({ orders }: Props) {
  return(
    <main className={styles.container}>
      <section className={styles.containerHeader}>
        <h1>Últimos pedidos</h1>
        <button>
          <RefreshCw size={24} color="#3fffa3" />
        </button>
      </section>

      <section className={styles.listOrders}>
        {orders.map((order) => (
          <button
            key={order.id}
            className={styles.orderItem}
          >
            <div className={styles.tag}></div>
            <span>Mesa {order.table}</span>
          </button>
        ))}
      </section>
    </main>
  )
}
```

Ln 21, Col 1 (280 selected) Spaces: 2 CRLF TypeScript JSX Prettier

Sujeito Pizza - A melhor pizzaria

localhost:300...

SujeitoPizza

Categoria Produto

Últimos pedidos

- Mesa 10
- Mesa 35
- Mesa 55

SujeitoPizza

[Categoria](#) [Produto](#) [\[→\]](#)

Últimos pedidos

Mesa 10

Mesa 35

Mesa 55

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file.
- Content:** The code is a TypeScript component named "Orders". It includes imports for "OrderProps" and "RefreshCw". The component returns a main container with a header section for "Últimos pedidos" and a button with a refresh icon. Below is a list of orders, each represented by a button with a key of "order.id" and a class of "orderItem". Each button contains a "tag" div and a "Mesa" span followed by the "table" value from the order object.

```
orders: OrderProps[]

export function Orders({ orders }: Props) {
  return(
    <main className={styles.container}>
      <section className={styles.containerHeader}>
        <h1>Últimos pedidos</h1>
        <button>
          <RefreshCw size={24} color="#3fffa3" />
        </button>
      </section>

      <section className={styles.listOrders}>
        {orders.map((order) => (
          <button
            key={order.id}
            className={styles.orderItem}
          >
            <div className={styles.tag}></div>
            <span>Mesa {order.table}</span>
          </button>
        ))}
      </section>
    </main>
  )
}
```

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localhost:300...
- Content:** The browser displays the application's homepage with the title "SujeitoPizza". It features a header with "Categoria" and "Produto" buttons. Below is a section titled "Últimos pedidos" showing two recent orders: "Mesa 35" and "Mesa 55".

File Edit View Ultimate Tools Help localhost:5432/pizzaria - Beekeeper Studio

pizzaria C + Query #1 orders [all] +

Filter ENTITIES 217 +

public

- _prisma_migrations
- categories
- items
- orders
- products
- users

information_schema

pg_catalog

| | id | equals | Enter Value | | | | | | | | | | | | | | | | |
|---|---------------------------------------|--------|-------------|-------|------------|--|--------|------|-------|-------|------|--------|------|------|--|----------------|--|--------------------------|--------------|
| | id | text | | table | int4(32,0) | | status | bool | | draft | bool | | name | text | | created_at... | | update_at | timestamp(3) |
| 1 | 0d0fef8f-9947-4e0a-94d2-dc2722b9cee0 | | | 55 | | | false | | false | | | (NULL) | | | | 2024-11-26 ... | | 2024-11-26 01:30:26.0... | |
| 2 | c9019c83-9a02-4152-a475-21f07e1e39... | | | 9 | | | true | | false | | | (NULL) | | | | 2024-11-26 ... | | 2024-11-26 15:36:31.8... | |
| 3 | ee42b24f-cdc5-48fd-87d3-1758a06afae3 | | | 35 | | | false | | false | | | Jorge | | | | 2024-11-29 ... | | 2024-11-29 02:28:31.9... | |

frontend

EXPLORER

FRONTEND

- .next
- node_modules
- public
- src
 - app
 - dashboard
 - category
 - components
 - button
 - header
 - orders
 - index.tsx
 - styles.module.scss
 - product
 - layout.tsx
 - page.tsx
 - fonts
 - signup
 - favicon.ico
 - globals.scss
 - layout.tsx
 - page.module.scss
 - page.tsx
 - lib
 - cookieClient.ts
 - cookieServer.ts
- OUTLINE
- TIMELINE

page.tsx index.tsx

src > app > dashboard > components > orders > index.tsx > Orders

```
orders: OrderProps[]

export function Orders({ orders }: Props) {
  return(
    <main className={styles.container}>
      <section className={styles.containerHeader}>
        <h1>Últimos pedidos</h1>
        <button>
          <RefreshCw size={24} color="#3ffffa3" />
        </button>
      </section>

      <section className={styles.listOrders}>
        {orders.map( order => (
          <button
            key={order.id}
            className={styles.orderItem}
          >
            <div className={styles.tag}></div>
            <span>Mesa {order.table}</span>
          </button>
        )));
      </section>
    </main>
  )
}
```

Ln 35, Col 2 Spaces: 2 UTF-8 CRLF TypeScript JSX ✓ Prettier

Sujeito Pizza - A melhor pizzaria

localhost:300...

SujeitoPizza

Categoria Produto [→]

Últimos pedidos

- Mesa 35
- Mesa 55

File Edit View Ultimate Tools Help localhost:5432/pizzaria - Beekeeper Studio

pizzaria C + Query #1 orders [all] +

Filter

ENTITIES 217 +

public

- _prisma_migrations
- categories
- items
- orders
- products
- users

information_schema

pg_catalog

| | id | table | status | draft | name | created_at... | update_at |
|---|---------------------------------------|-------|--------|-------|--------|----------------|--------------------------|
| 1 | 0d0fef8f-9947-4e0a-94d2-dc2722b9cee0 | 55 | false | false | (NULL) | 2024-11-26 ... | 2024-11-26 01:30:26.0... |
| 2 | c9019c83-9a02-4152-a475-21f07e1e39... | 9 | true | false | (NULL) | 2024-11-26 ... | 2024-11-26 15:36:31.8... |
| 3 | ee42b24f-cdc5-48fd-87d3-1758a06afae3 | 35 | false | false | Jorge | 2024-11-29 ... | 2024-11-29 02:28:31.9... |

File Edit View Ultimate Tools Help localhost:5432/pizzaria - Beekeeper Studio

pizzaria C + Query #1 orders [all] +

Filter

ENTITIES 217 +

public

- _prisma_migrations
- categories
- items
- orders
- products
- users

information_schema

pg_catalog

| | id | equals | Enter Value | | | | |
|---|---------------------------------------|--------|------------------------|-------|--------|----------------|--------------------------|
| 1 | 0d0fef8f-9947-4e0a-94d2-dc2722b9cee0 | 55 | false | false | (NULL) | 2024-11-26 ... | 2024-11-26 01:30:26.0... |
| 2 | c9019c83-9a02-4152-a475-21f07e1e39... | 9 | true | false | (NULL) | 2024-11-26 ... | 2024-11-26 15:36:31.8... |
| 3 | ee42b24f-cdc5-48fd-87d3-1758a06afae3 | 35 | false false true | false | Jorge | 2024-11-29 ... | 2024-11-29 02:28:31.9... |

File Edit View Ultimate Tools Help localhost:5432/pizzaria - Beekeeper Studio

pizzaria C + Query #1 orders [all] +

Filter

ENTITIES 217 +

public

- _prisma_migrations
- categories
- items
- orders
- products
- users

information_schema

pg_catalog

id equals Enter Value

| | id | table | status | draft | name | created_at... | update_at |
|---|---------------------------------------|-------|--------|-------|--------|----------------|--------------------------|
| 1 | 0d0fef8f-9947-4e0a-94d2-dc2722b9cee0 | 55 | false | false | (NULL) | 2024-11-26 ... | 2024-11-26 01:30:26.0... |
| 2 | c9019c83-9a02-4152-a475-21f07e1e39... | 9 | true | false | (NULL) | 2024-11-26 ... | 2024-11-26 15:36:31.8... |
| 3 | ee42b24f-cdc5-48fd-87d3-1758a06afae3 | 35 | true | false | Jorge | 2024-11-29 ... | 2024-11-29 02:28:31.9... |

File Edit View Ultimate Tools Help localhost:5432/pizzaria - Beekeeper Studio

pizzaria C + Query #1 orders [all] +

Filter

ENTITIES 217 +

public

- _prisma_migrations
- categories
- items
- orders
- products
- users

information_schema

pg_catalog

| | id | equals | Enter Value | | | | | | | | | | | | | | | | | | | |
|---|---------------------------------------|--------|-------------|-------|------------|--|--------|------|--|-------|------|--|--------|------|------|--|--|----------------|--|--------------------------|--------------|--|
| | id | text | | table | int4(32,0) | | status | bool | | draft | bool | | | name | text | | | created_at... | | update_at | timestamp(3) | |
| 1 | 0d0fef8f-9947-4e0a-94d2-dc2722b9cee0 | 55 | | | | | false | | | false | | | (NULL) | | | | | 2024-11-26 ... | | 2024-11-26 01:30:26.0... | | |
| 2 | c9019c83-9a02-4152-a475-21f07e1e39... | 9 | | | | | true | | | false | | | (NULL) | | | | | 2024-11-26 ... | | 2024-11-26 15:36:31.8... | | |
| 3 | ee42b24f-cdc5-48fd-87d3-1758a06afae3 | 35 | | | | | true | | | false | | | Jorge | | | | | 2024-11-29 ... | | 2024-11-29 02:28:31.9... | | |

frontend

EXPLORER

FRONTEND

- .next
- node_modules
- public
- src
 - app
 - dashboard
 - category
 - components
 - button
 - header
 - orders
 - index.tsx
 - styles.module.scss
 - product
 - layout.tsx
 - page.tsx
 - fonts
 - signup
 - favicon.ico
 - globals.scss
 - layout.tsx
 - page.module.scss
 - page.tsx
 - lib
 - cookieClient.ts
 - cookieServer.ts

OUTLINE

TIMELINE

page.tsx index.tsx

src > app > dashboard > components > orders > index.tsx > Orders

```
orders: OrderProps[]

export function Orders({ orders }: Props) {
  return(
    <main className={styles.container}>
      <section className={styles.containerHeader}>
        <h1>Últimos pedidos</h1>
        <button>
          <RefreshCw size={24} color="#3fffa3" />
        </button>
      </section>

      <section className={styles.listOrders}>
        {orders.map( order => (
          <button
            key={order.id}
            className={styles.orderItem}
            >
              <div className={styles.tag}></div>
              <span>Mesa {order.table}</span>
            </button>
        )));
      </section>
    </main>
  )
}
```

Ln 35, Col 2 Spaces: 2 UTF-8 CRLF TypeScript JSX Prettier

Sujeito Pizza - A melhor pizzaria

localhost:300...

SujeitoPizza

Categoria Produto

Últimos pedidos

- Mesa 35
- Mesa 55

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file, showing TypeScript code for rendering orders.
- Output:** Shows the command line interface with various status messages.

```
page.tsx index.tsx
src > app > dashboard > components > orders > index.tsx > Orders
orders: OrderProps[]

export function Orders({ orders }: Props) {
  return(
    <main className={styles.container}>
      <section className={styles.containerHeader}>
        <h1>Últimos pedidos</h1>
        <button>
          <RefreshCw size={24} color="#3fffa3" />
        </button>
      </section>

      <section className={styles.listOrders}>
        {orders.map( order => (
          <button
            key={order.id}
            className={styles.orderItem}
          >
            <div className={styles.tag}></div>
            <span>Mesa {order.table}</span>
          </button>
        )));
      </section>
    </main>
  )
}
```

Browser Preview:

The browser window displays the application "Sujeito Pizza - A melhor pizzaria" at localhost:3001. The page shows the title "SujeitoPizza" and a section titled "Últimos pedidos" with a button labeled "Mesa 55".

File Edit View Ultimate Tools Help localhost:5432/pizzaria - Beekeeper Studio

pizzaria C + Query #1 orders [all] +

Filter ENTITIES 217 +

public

- _prisma_migrations
- categories
- items
- orders
- products
- users

information_schema

pg_catalog

| | id | table | status | draft | name | created_at... | update_at |
|---|---------------------------------------|-------|--------|-------|--------|----------------|--------------------------|
| 1 | 0d0fef8f-9947-4e0a-94d2-dc2722b9cee0 | 55 | false | false | (NULL) | 2024-11-26 ... | 2024-11-26 01:30:26.0... |
| 2 | c9019c83-9a02-4152-a475-21f07e1e39... | 9 | true | false | (NULL) | 2024-11-26 ... | 2024-11-26 15:36:31.8... |
| 3 | ee42b24f-cdc5-48fd-87d3-1758a06afae3 | 35 | true | false | Jorge | 2024-11-29 ... | 2024-11-29 02:28:31.9... |

File Edit View Ultimate Tools Help localhost:5432/pizzaria - Beekeeper Studio

pizzaria C + Query #1 orders [all] x +

Filter ENTITIES 217 +

public

- _prisma_migrations
- categories
- items
- orders
- products
- users

information_schema

pg_catalog

| | id | table | status | draft | name | created_at... | update_at |
|---|---------------------------------------|-------|--------|-------|--------|----------------|--------------------------|
| 1 | 0d0fef8f-9947-4e0a-94d2-dc2722b9cee0 | 55 | false | false | (NULL) | 2024-11-26 ... | 2024-11-26 01:30:26.0... |
| 2 | c9019c83-9a02-4152-a475-21f07e1e39... | 9 | true | false | (NULL) | 2024-11-26 ... | 2024-11-26 15:36:31.8... |
| 3 | ee42b24f-cdc5-48fd-87d3-1758a06afae3 | 35 | true | false | Jorge | 2024-11-29 ... | 2024-11-29 02:28:31.9... |

1 > C +

File Edit View Ultimate Tools Help localhost:5432/pizzaria - Beekeeper Studio

pizzaria C + Query #1 orders [all] +

Filter

ENTITIES 217 +

public

- _prisma_migrations
- categories
- items
- orders
- products
- users

information_schema

pg_catalog

| | id | equals | Enter Value | | | | |
|---|---------------------------------------|--------|-------------|-------|--------|----------------|--------------------------|
| 1 | 0d0fef8f-9947-4e0a-94d2-dc2722b9cee0 | 55 | false | false | (NULL) | 2024-11-26 ... | 2024-11-26 01:30:26.0... |
| 2 | c9019c83-9a02-4152-a475-21f07e1e39... | 9 | true | false | (NULL) | 2024-11-26 ... | 2024-11-26 15:36:31.8... |
| 3 | ee42b24f-cdc5-48fd-87d3-1758a06afae3 | 35 | false | false | Jorge | 2024-11-29 ... | 2024-11-29 02:28:31.9... |

File Edit View Ultimate Tools Help localhost:5432/pizzaria - Beekeeper Studio

pizzaria C + Query #1 orders [all] +

Filter

ENTITIES 217 +

- public
 - _prisma_migrations
 - categories
 - items
 - orders
 - products
 - users
- information_schema
- pg_catalog

| | id | equals | Enter Value | | | | | | | | | | | | | | | | | |
|---|---------------------------------------|--------|-------------|-------|------------|--|--------|------|--|-------|------|--|--------|------|--|----------------|--|--------------------------|--------------|--|
| | id | text | | table | int4(32,0) | | status | bool | | draft | bool | | name | text | | created_at... | | update_at | timestamp(3) | |
| 1 | 0d0fef8f-9947-4e0a-94d2-dc2722b9cee0 | 55 | | | | | false | | | false | | | (NULL) | | | 2024-11-26 ... | | 2024-11-26 01:30:26.0... | | |
| 2 | c9019c83-9a02-4152-a475-21f07e1e39... | 9 | | | | | true | | | false | | | (NULL) | | | 2024-11-26 ... | | 2024-11-26 15:36:31.8... | | |
| 3 | ee42b24f-cdc5-48fd-87d3-1758a06afae3 | 35 | | | | | false | | | false | | | Jorge | | | 2024-11-29 ... | | 2024-11-29 02:28:31.9... | | |

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file, showing TypeScript code for rendering orders.
- Output:** Shows the file path: "src > app > dashboard > components > orders > index.tsx".
- Editor:** The code defines a component named "Orders" that maps over an array of "orders" to render individual order items.

```
6  orders: OrderProps[]  
7 }  
8  
9 export function Orders({ orders }: Props){  
10 return(  
11     <main className={styles.container}>  
12         <section className={styles.containerHeader}>  
13             <h1>Últimos pedidos</h1>  
14             <button>  
15                 <RefreshCw size={24} color="#3fffa3" />  
16             </button>  
17         </section>  
18  
19         <section className={styles.listOrders}>  
20             {orders.map( order => (  
21                 <button  
22                     key={order.id}  
23                     className={styles.orderItem}  
24                     >  
25                         <div className={styles.tag}></div>  
26                         <span>Mesa {order.table}</span>  
27                     </button>  
28             ))}  
29         </section>  
30  
31     </main>  
32  
33 }  
34  
35 }
```

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localhost:300...
- Content:** Displays the heading "Últimos pedidos" and a button labeled "Mesa 55".

The screenshot shows the VS Code interface with the following details:

- Explorer View:** Shows the project structure under "FRONTEND".
- Search Bar:** Contains the text "frontend".
- Editor:** Displays the "index.tsx" file content. The code is a functional component named "Orders" that renders a header section with a refresh button and a list of orders. The list items include a key prop and a span element with the text "Mesa".
- Bottom Status Bar:** Shows "Ln 35, Col 2" and other file-related information.

```
page.tsx index.tsx
src > app > dashboard > components > orders > index.tsx > Orders
orders: OrderProps[]

export function Orders({ orders }: Props) {
  return(
    <main className={styles.container}>
      <section className={styles.containerHeader}>
        <h1>Últimos pedidos</h1>
        <button>
          <RefreshCw size={24} color="#3fffa3" />
        </button>
      </section>

      <section className={styles.listOrders}>
        {orders.map( order => (
          <button
            key={order.id}
            className={styles.orderItem}
          >
            <div className={styles.tag}></div>
            <span>Mesa {order.table}</span>
          </button>
        )));
      </section>
    </main>
  )
}
```

The screenshot shows a web browser window displaying the "Sujeito Pizza" website at "localhost:300...". The page features a header with the brand name and navigation links for "Categoria" and "Produto". Below the header, there is a section titled "Últimos pedidos" (Last orders) which lists two entries: "Mesa 35" and "Mesa 55".

SujeitoPizza

Categoria Produto

Últimos pedidos

Mesa 35

Mesa 55

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file, showing TypeScript code for a component named "Orders".
- Code:**

```
src > app > dashboard > components > orders > index.tsx > Orders
6   orders: OrderProps[]
7 }
8
9 export function Orders({ orders }: Props){
10   return(
11     <main className={styles.container}>
12
13       <section className={styles.containerHeader}>
14         <h1>Últimos pedidos</h1>
15         <button>
16           <RefreshCw size={24} color="#3ffffa3" />
17         </button>
18       </section>
19
20       <section className={styles.listOrders}>
21         {orders.map( order => (
22           <button
23             key={order.id}
24             className={styles.orderItem}
25           >
26             <div className={styles.tag}></div>
27             <span>Mesa {order.table}</span>
28           </button>
29         )));
30       </section>
31
32     </main>
33   )
34 }
```
- Bottom Status Bar:** Shows file path, line/col, spaces, encoding, and Prettier status.

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localhost:300...
- Content:** Displays a dark-themed form titled "Novo produto" for adding a new product. It includes fields for "bebidas", a search bar ("Digite o nome do produto..."), a price input ("Preço do produto..."), and a description input ("Digite a descrição do produto..."). A green "Cadastrar produto" button is at the bottom.

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file, showing TypeScript code for a component named "Orders".
- Code:** The code uses React components and hooks like "orders" and "OrderProps". It includes sections for "Últimos pedidos" and a list of orders.
- Status Bar:** Shows file statistics: Ln 35, Col 2, Spaces: 2, UTF-8, CRLF, and build status: TypeScript JSX, Prettier.

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localhost:3000
- Content:** Displays a dark-themed web page titled "SujeitoPizza" with a header "Nova Categoria". It has a search bar labeled "Nome da categoria, ex: Pizzas" and a green "Cadastrar" button.
- Status Bar:** Shows the current URL: localhost:3000/dashboard/category.

frontend

EXPLORER

FRONTEND

- .next
- node_modules
- public
- src
 - app
 - dashboard
 - category
 - components
 - button
 - header
 - orders
 - index.tsx
 - styles.module.scss
 - product
 - layout.tsx
 - page.tsx
 - fonts
 - signup
 - favicon.ico
 - globals.scss
 - layout.tsx
 - page.module.scss
 - page.tsx
 - lib
 - cookieClient.ts
 - cookieServer.ts

OUTLINE

TIMELINE

page.tsx index.tsx

src > app > dashboard > components > orders > index.tsx > Orders

```
orders: OrderProps[]

export function Orders({ orders }: Props) {
  return(
    <main className={styles.container}>

      <section className={styles.containerHeader}>
        <h1>Últimos pedidos</h1>
        <button>
          <RefreshCw size={24} color="#3fffa3" />
        </button>
      </section>

      <section className={styles.listOrders}>
        {orders.map( order => (
          <button
            key={order.id}
            className={styles.orderItem}
            >
              <div className={styles.tag}></div>
              <span>Mesa {order.table}</span>
            </button>
        )));
      </section>
    </main>
  )
}
```

Ln 35, Col 2 Spaces: 2 UTF-8 CRLF TypeScript JSX ✓ Prettier

Sujeito Pizza - A melhor pizzaria

localhost:300...

SujeitoPizza

Categoria Produto

Últimos pedidos

- Mesa 35
- Mesa 55

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file, showing TypeScript code for rendering a list of orders.
- Code:**

```
src > app > dashboard > components > orders > index.tsx > Orders
6   orders: OrderProps[]
7 }
8
9 export function Orders({ orders }: Props){
10   return(
11     <main className={styles.container}>
12
13       <section className={styles.containerHeader}>
14         <h1>Últimos pedidos</h1>
15         <button>
16           <RefreshCw size={24} color="#3fffa3" />
17         </button>
18       </section>
19
20       <section className={styles.listOrders}>
21         {orders.map( order => (
22           <button
23             key={order.id}
24             className={styles.orderItem}
25             >
26               <div className={styles.tag}></div>
27               <span>Mesa {order.table}</span>
28             </button>
29         ))}
30
31       </section>
32
33     </main>
34   )
35 }
```

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localhost:3000
- Content:** Displays the application's login page with fields for email and password, and links for forgot password and register.
- Status:** A success message at the bottom right indicates "Logout Realizado com Sucesso!".

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file, showing TypeScript code for rendering a list of orders.
- Output:** Shows the file path: "src > app > dashboard > components > orders > index.tsx".
- Editor:** The code defines a component named "Orders" that maps over an array of "orders" to render individual order items.

```
6  orders: OrderProps[]  
7 }  
8  
9 export function Orders({ orders }: Props){  
10   return(  
11     <main className={styles.container}>  
12       <section className={styles.containerHeader}>  
13         <h1>Últimos pedidos</h1>  
14         <button>  
15           <RefreshCw size={24} color="#3fffa3" />  
16         </button>  
17       </section>  
18  
19       <section className={styles.listOrders}>  
20         {orders.map( order => (  
21           <button  
22             key={order.id}  
23             className={styles.orderItem}  
24             >  
25               <div className={styles.tag}></div>  
26               <span>Mesa {order.table}</span>  
27             </button>  
28         ))}  
29       </section>  
30     </main>  
31   )  
32 }  
33 }  
34 }  
35 }
```

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- Content:** Displays the application's login screen with the title "SujeitoPizza". It includes fields for email ("regina@teste.com") and password ("----"), a "Acessar" button, and a link to "Cadastrar-se".

```
src > app > dashboard > components > orders > index.tsx > Orders
orders: OrderProps[]

export function Orders({ orders }: Props) {
  return(
    <main className={styles.container}>
      <section className={styles.containerHeader}>
        <h1>Últimos pedidos</h1>
        <button>
          <RefreshCw size={24} color="#3ffffa3" />
        </button>
      </section>

      <section className={styles.listOrders}>
        {orders.map( order => (
          <button
            key={order.id}
            className={styles.orderItem}
          >
            <div className={styles.tag}></div>
            <span>Mesa {order.table}</span>
          </button>
        ))}
      </section>
    </main>
  )
}
```

A screenshot of a web browser window. The address bar shows the URL 'localhos...'. The main content area features the logo 'SujeitoPizza' in a large, bold, black font. To the right of the logo are three buttons labeled 'Categoria', 'Produto', and a right-pointing arrow. Below the logo, the text 'Últimos pedidos' is displayed in white, followed by a green circular arrow icon. Two dark blue rectangular boxes list the latest orders: 'Mesa 35' and 'Mesa 55'. The background of the page is dark.

UC: Projetos

Assunto:

Projeto Pizzaria – Frontend_02 (NextJS)

Sistema Pizzaria Web

58 – Criando Modal de Pedido

```
npm run dev

Microsoft Windows [versão 10.0.19045.5131]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\Users\Dev>cd Desktop\Pizzaria\backend

C:\Users\Dev\Desktop\Pizzaria\backend>npm run dev

> backend@1.0.0 dev
> ts-node-dev --transpile-only src/server.ts

[INFO] 18:47:21 ts-node-dev ver. 2.0.0 (using ts-node ver. 10.9.2, typescript ver. 5.6.3)
Servidor online!!!!
```

```
Import trace for requested module:  
.src/app/dashboard/product/components/form/styles.module.scss  
.src/app/dashboard/product/components/form/index.tsx
```

```
.src/app/page.module.scss
```

```
Deprecation The legacy JS API is deprecated and will be removed in Dart Sass 2.0.0.
```

```
More info: https://sass-lang.com/d/legacy-js-api
```

```
Import trace for requested module:
```

```
.src/app/page.module.scss
```

```
.src/app/page.tsx
```

```
GET /dashboard/category 200 in 1113ms
```

```
true
```

```
GET /dashboard 200 in 50ms
```

```
GET / 200 in 31ms
```

```
{
```

```
  id: 'c47ca9e5-2013-4fb9-821f-8ce9bd8ee2e5',
```

```
  name: 'Regina Dias',
```

```
  email: 'regina@teste.com',
```

```
  token: 'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJJuYW1lIjoiUmVnaW5hIERpYXMiLCJlbWFpbCI6InJlZ2luYUB0ZXN0ZS5jb20iLCJpYXQiOjE3MzMyNjQ4NjEsImV4cCI6MTczNTg1Njg2MSwiic3ViIjoiYzQ3Y2E5ZTUtMjAxMy00ZmI5LTgyMjYtOGNlOWJkOGVlMmU1In0._z10k4ZAEvyLrn-qfJeh7uh2-gCL16JETZsxsuIKpVU'
```

```
}
```

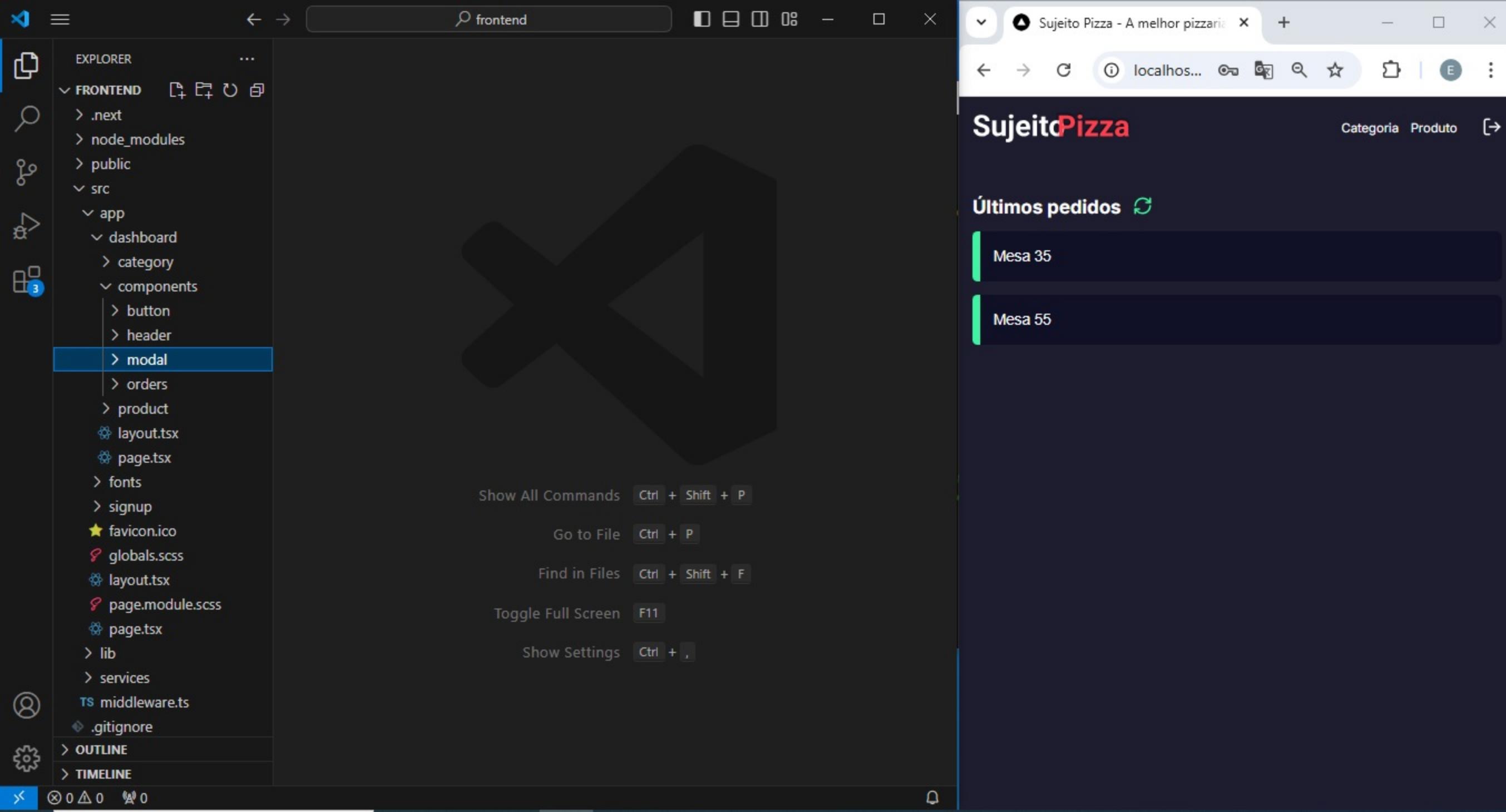
```
true
```

```
GET /dashboard 200 in 34ms
```

```
POST / 303 in 126ms
```

```
true
```

```
GET /dashboard 200 in 23ms
```



The image shows a developer's workspace with a code editor and a web browser side-by-side.

Code Editor (VS Code):

- Explorer:** Shows the project structure under `FRONTEND`.
 - `src` contains `app`, `public`, and `node_modules`.
 - `app` contains `category`, `button`, `header`, and `modal`.
 - `modal` contains `index.tsx` (selected), `orders`, `product`, `layout.tsx`, `page.tsx`, `fonts`, `signup`, `favicon.ico`, `globals.scss`, `layout.tsx`, `page.module.scss`, `page.tsx`, `lib`, `services`, `middleware.ts`, and `OUTLINE`, `TIMELINE`.
- Status Bar:** Shows file statistics: `Ln 1, Col 1`, `Spaces: 4`, `UTF-8`, `CRLF`, `TypeScript JSX`, `Prettier`, and `0 0 0 0 0`.

Browser:

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localhos... (highlighted)
- Content:** Displays the "Últimos pedidos" (Last orders) section, showing two entries:
 - Mesa 35
 - Mesa 55
- Header:** Shows `Categoria` and `Produto` buttons.

The image shows a developer's environment with a code editor and a browser preview side-by-side.

Code Editor (VS Code):

- Explorer:** Shows the project structure under `FRONTEND`.
 - `src` contains `app`, which has `category`, `components` (containing `button` and `header`), and `modal`. `modal` contains `index.tsx` and `styles.module.scss`.
 - `public`
 - `node_modules`
 - `.next`
- Search:** Shows results for `frontend` across the project.
- File:** The file `styles.module.scss` is currently open in the editor.
- Bottom Status Bar:** Shows file path (`src > app > dashboard > components > modal > styles.module.scss`), line 1, column 1, and other status indicators.

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- Header:** Shows `Categoria` and `Produto` buttons.
- Section:** **Últimos pedidos** (Last orders) with a refresh icon.
- List:** Shows two recent orders:
 - Mesa 35
 - Mesa 55

The image shows a developer's workspace with a code editor and a browser preview side-by-side.

Code Editor (left):

- Explorer:** Shows the project structure under "FRONTEND".
- index.tsx:** Contains the following TypeScript code:

```
1 import styles from './styles.module.scss'
2
3 export function Modalorder(){
4     return(
5         <div>
6             <h1> TESTE </h1>
7         </div>
8     )
9 }
```
- styles.module.scss:** Placeholder file.

Browser Preview (right):

The browser displays the application "Sujeito Pizza - A melhor pizzaria".

- Header:** "SujeitoPizza" with navigation links "Categoria" and "Produto".
- Section:** "Últimos pedidos" (Last orders) with two items:
 - Mesa 35
 - Mesa 55

At the bottom of the screen, there are status indicators: "Ln 1, Col 1 (158 selected)", "Spaces: 4", "UTF-8", "CRLF", "TypeScript JSX", "Prettier", and a bell icon.

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- Search:** Shows search results.
- File Explorer:** Shows files in the current directory.
- Outline:** Shows code structure.
- Timeline:** Shows project history.
- Editor:** Displays the file `src/app/dashboard/components/orders/index.tsx`. The code defines a component named `Orders` that displays the last orders. It uses `OrderProps` as a prop type and `RefreshCw` from the `@material-ui/icons` package.
- Status Bar:** Shows file statistics (2 selections, 5 characters selected), encoding (UTF-8), line endings (CRLF), and Prettier status.

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localhos...
- Content:** The page displays the title "SujeitoPizza" and a section titled "Últimos pedidos" (Last orders) with two items: "Mesa 35" and "Mesa 55".
- Header:** Includes links for "Categoria" and "Produto".

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file, showing TypeScript code for a component named "Orders".
- Imports:** The code imports "styles.module.scss", "RefreshCw" from "lucide-react", "OrderProps" from "@lib/order.type", and "Modalorder" from "../modal".
- Component Structure:** The component returns a main container with a header section containing a button with a RefreshCw icon, followed by a list of orders.
- List Section:** The list iterates over "orders" and creates a button for each order, displaying the table number (e.g., Mesa 35, Mesa 55).

Browser Preview:

The browser displays the application "SujeitoPizza - A melhor pizzaria" at localhost:3001. The page shows the title "SujeitoPizza" and a section titled "Últimos pedidos" (Last orders) which lists "Mesa 35" and "Mesa 55".

```
import styles from './styles.module.scss'
import { RefreshCw } from 'lucide-react'
import { OrderProps } from '@/lib/order.type'
import { Modalorder } from '../modal'

interface Props{
    orders: OrderProps[]
}

export function Orders({ orders }: Props){
    return(
        <main className={styles.container}>
            <section className={styles.containerHeader}>
                <h1>Últimos pedidos</h1>
                <button>
                    <RefreshCw size={24} color="#3fffa3" />
                </button>
            </section>

            <section className={styles.listOrders}>
                {orders.map( order => (
                    <button
                        key={order.id}
                        className={styles.orderItem}
                    >
                        <div className={styles.tag}></div>
                        <span>Mesa {order.table}</span>
                    </button>
                )))
            </section>
        </main>
    )
}
```

frontend

EXPLORER

FRONTEND

- > .next
- > node_modules
- > public
- src
 - app
 - dashboard
 - > category
 - > components
 - > button
 - > header
 - > modal
 - > orders
 - index.tsx
 - styles.module.scss
 - > product
 - layout.tsx
 - page.tsx
 - > fonts
 - > signup
 - favicon.ico
 - globals.scss
 - layout.tsx
 - page.module.scss
 - page.tsx
 - > lib
 - > services

OUTLINE

TIMELINE

index.tsx ..\modal index.tsx ...orders styles.module.scss

src > app > dashboard > components > orders > index.tsx > Orders

```
12    <>
13      <main className={styles.container}>
14        <section className={styles.containerHeader}>
15          <h1>Últimos pedidos</h1>
16          <button>
17            <RefreshCw size={24} color="#3ffffa3" />
18          </button>
19        </section>
20
21        <section className={styles.listOrders}>
22          {orders.map( order => (
23            <button
24              key={order.id}
25              className={styles.orderItem}
26            >
27              <div className={styles.tag}></div>
28              <span>Mesa {order.table}</span>
29            </button>
30          ))}
31        </section>
32
33      </main>
34
35      <Modalorder/>
36
37    </>
38
39  }
40 }
```

Ln 37, Col 5 (13 selected) Spaces: 2 UTF-8 CRLF TypeScript JSX Prettier

Sujeito Pizza - A melhor pizzaria

localhost:300...

SujeitoPizza

Categoria Produto []

Últimos pedidos

- Mesa 35
- Mesa 55

TESTE

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file, showing TypeScript code for rendering a dashboard component.
- Search:** "frontend" is the search term.
- Outline:** Available at the bottom.
- Timeline:** Available at the bottom.

```
12  <>
13  <main className={styles.container}>
14
15  <section className={styles.containerHeader}>
16    <h1>Últimos pedidos</h1>
17    <button>
18      <RefreshCw size={24} color="#3ffffa3" />
19    </button>
20  </section>
21
22  <section className={styles.listOrders}>
23    {orders.map( order => (
24      <button
25        key={order.id}
26        className={styles.orderItem}
27        >
28          <div className={styles.tag}></div>
29          <span>Mesa {order.table}</span>
30        </button>
31      ))}
32
33  </section>
34
35  </main>
36
37  <Modalorder/>
38
39  )
40 }
```

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localhost:300...
- Content:** Displays the application's UI, including a header with "SujeitoPizza" and navigation links, a section titled "Últimos pedidos" showing recent orders for "Mesa 35" and "Mesa 55", and a large blue button labeled "TESTE".

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file.
- Content:** The code defines a function "Modalorder" that returns a simple HTML structure with an

element containing the text "TESTE".

```
1 import styles from './styles.module.scss'
2
3 export function Modalorder(){
4     return(
5         <div>
6             <h1> TESTE </h1>
7         </div>
8     )
9 }
```

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localhost:300...
- Header:** Categoriá Produto
- Section:** Últimos pedidos
- Items:** Mesa 35, Mesa 55
- Text:** TESTE

The image shows a developer's environment with two main windows: a code editor in the foreground and a browser preview in the background.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file.
- Content:** The code defines a modal component named "Modalorder".

```
1 import styles from './styles.module.scss'
2
3 export function Modalorder(){
4     return(
5         <dialog>
6             <h1> TESTE </h1>
7         </dialog>
8     )
9 }
```

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localhost:300...
- Content:** Displays the "Últimos pedidos" section with two items:
 - Mesa 35
 - Mesa 55

Both windows have standard operating system window controls (minimize, maximize, close) at the top.

The image shows a developer's environment with two main windows: a code editor in the foreground and a browser preview in the background.

Code Editor (VS Code):

- Explorer:** Shows the project structure under `FRONTEND`. The `src` folder contains `app`, which further contains `dashboard`, `category`, `components`, `button`, `header`, and `modal`.
- File Explorer:** Shows files like `index.tsx`, `styles.module.scss`, `orders/index.tsx`, `products/index.tsx`, `layout.tsx`, `page.tsx`, `fonts`, `signup`, `favicon.ico`, `globals.scss`, `layout.module.scss`, `page.module.scss`, and `page.tsx`.
- Editor:** Displays the `index.tsx` file content:

```
1 import styles from './styles.module.scss'
2
3 export function Modalorder(){
4     return(
5         <dialog className={styles.dialogContainer}>
6             <h1> TESTE </h1>
7         </dialog>
8     )
9 }
```

Browser Preview:

The browser window displays the application "Sujeito Pizza - A melhor pizzaria". The page shows the header "SujeitoPizza" and a section titled "Últimos pedidos" (Last orders) with two items: "Mesa 35" and "Mesa 55".

Bottom Status Bar:

Shows the current file is `index.tsx`, with 34 selected characters at Ln 5, Col 51. It also indicates `Spaces: 4`, `UTF-8`, `CRLF`, `TypeScript JSX`, and `Prettier` are enabled.

The image shows a developer's environment with two main windows: a code editor in the foreground and a browser preview in the background.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file.
- Content:** The code defines a function "Modalorder" that returns a modal dialog with a heading "TESTE".

```
import styles from './styles.module.scss'

export function Modalorder(){
    return(
        <dialog className={styles.dialogContainer}>
            <h1>TESTE</h1>
        </dialog>
    )
}
```

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localhost:300...
- Content:** Displays the "Últimos pedidos" section with two items: "Mesa 35" and "Mesa 55".

The image shows a developer's environment with two main windows: a code editor and a web browser.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file.
- Content:** The code defines a function "Modalorder" that returns a modal dialog with a heading "TESTE".

```
index.tsx
import styles from './styles.module.scss'

export function Modalorder(){
    return(
        <dialog className={styles.dialogContainer}>
            <h1>TESTE</h1>
        </dialog>
    )
}
```

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localhost:300...
- Content:** Displays the "Últimos pedidos" section with two items: "Mesa 35" and "Mesa 55".

The image shows a developer's workspace split between a code editor and a web browser.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file, containing the following TypeScript code:

```
import styles from './styles.module.scss'

export function Modalorder(){
    return(
        <dialog className={styles.dialogContainer}>
            </dialog>
    )
}
```
- Search:** A search bar at the top of the editor.
- Icons:** Standard VS Code icons for file operations.

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- Address Bar:** localhost:300...
- Content:** Displays the "Últimos pedidos" (Last orders) section of the application, listing "Mesa 35" and "Mesa 55".
- Header:** "SujeitoPizza" logo, "Categoria", "Produto", and a search icon.

Bottom Status Bar:

- Ln 6, Col 13
- Spaces: 4
- UTF-8
- CRLF
- TypeScript JSX
- Prettier

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is open in the editor.
- Content:** The code is a modal component named "Modalorder". It imports styles from "./styles.module.scss" and uses "lucide-react" for icons. The component returns a dialog with a header containing a close button (X icon) and a section for order details. It lists two items: "Pizza catupiry" and "Pizza de frango com catupiry, borda recheada".
- Status Bar:** Shows line 40, column 2 (1136 selected), spaces: 4, encoding: UTF-8, CRLF, and Prettier status.

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- Header:** Categories and Products.
- Section:** Últimos pedidos (Last orders).
- Items:** Mesa 35 and Mesa 55.

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** index.tsx (selected), styles.module.scss
- Content:** TypeScript code for a modal component. It imports styles from "styles.module.scss" and "lucide-react". The component returns a dialog with a close button and an article containing details about a pizza order. The code uses class names defined in "styles.module.scss".
- Status Bar:** Lines 40, Col 2 (1136 selected), Spaces: 4, UTF-8, CRLF, TypeScript JSX, Prettier.

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- Header:** Categoria, Produto, [Entrar]
- Section:** Últimos pedidos
- Content:** Mesa 35, Mesa 55

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file.
- Content:** The code is a modal component. The last few lines are:

```
          Concluir pedido
      </button>
    </article>
  </section>
</dialog>
)
```
- Status Bar:** Shows "Ln 40, Col 2 (1136 selected)" and "Spaces: 4".

Browser Preview:

- Title:** "Sujeito Pizza - A melhor pizzaria"
- Header:** "SujeitoPizza" logo, "Categoria", "Produto", and a search bar.
- Section:** "Últimos pedidos" with a refresh icon.
- List:** Two items: "Mesa 35" and "Mesa 55".

Bottom Status Bar: Shows "Ln 40, Col 2 (1136 selected)", "Spaces: 4", "UTF-8", "CRLF", "TypeScript JSX", "Prettier", and a lock icon.

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file.
- File:** "styles.module.scss" is also open.
- Content:** The "styles.module.scss" file contains SCSS code for a modal dialog. It includes rules for ".dialogContainer" and ".dialogContent". The ".dialogContainer" rule sets a fixed position, dimensions, background, and z-index. The ".dialogContent" rule sets a background color, margin, padding, border-radius, width, max-width, position, and color.
- Status Bar:** Shows line 32, column 4 (613 selected), spaces: 4, and encoding: UTF-8.

Browser Preview:

- Title:** "Sujeito Pizza - A melhor pizzaria"
- Content:** A modal dialog titled "Detalhes do pedido" (Order Details) for "Mesa 36".
 - Item 1:** "1 - Pizza catupiry" (Chicken pizza with catupiry, filled border). Description: "Pizza de frango com catupiry, borda recheada".
 - Item 3:** "3 - Pizza calabresa" (Calabrese pizza). Description: "Pizza de frango com catupiry, borda recheada".
- Buttons:** A green "Concluir pedido" (Finish order) button.

The image shows a developer's workspace with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file.
- File:** "styles.module.scss" is also open.
- Content:** The "styles.module.scss" file contains SCSS code for a modal component. It includes rules for ".table", ".dialogBack", ".item", ".description", and ".buttonOrder".

```
.table{  
  background-color: var(--dark-900);  
  padding: 8px;  
  border-radius: 4px;  
}  
  
.dialogBack{  
  background-color: transparent;  
  border: 0;  
}  
  
.item{  
  display: flex;  
  flex-direction: column;  
  margin: 16px 0;  
  font-size: 16px;  
}  
  
.description{  
  color: #CCC;  
  margin-top: 4px;  
}  
  
.buttonOrder{  
  background-color: var(--green-900);  
  padding: 8px;  
  border-radius: 4px;  
  border: 0;  
  font-weight: bold;  
}
```

Browser Preview:

The browser shows a modal window titled "Detalhes do pedido" (Details of the order) for "Mesa 36".

- Items:**
 - 1 - Pizza catupiry: Pizza de frango com catupiry, borda recheada
 - 3 - Pizza calabresa: Pizza de frango com catupiry, borda recheada
- Buttons:** A green "Concluir pedido" (Finish order) button.

UC: Projetos

Assunto:

Projeto Pizzaria – Frontend_02 (NextJS)

Sistema Pizzaria Web

59 – Funcionalidade do Modal

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is open in the editor.
- Content:** The code is a React component named "Orders". It displays the last orders with a modal for details.

```
<main className={styles.container}>
  <section className={styles.containerHeader}>
    <h1>Últimos pedidos</h1>
    <button>
      <RefreshCw size={24} color="#3ffffa3" />
    </button>
  </section>

  <section className={styles.listOrders}>
    {orders.map( order => (
      <button
        key={order.id}
        className={styles.orderItem}
      >
        <div className={styles.tag}></div>
        <span>Mesa {order.table}</span>
      </button>
    ))}
  </section>
</main>

<Modalorder/>
```

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- Content:** A modal window titled "Detalhes do pedido" for "Mesa 36". It lists two pizzas:
 - 1 - Pizza catupiry: Pizza de frango com catupiry, borda recheada
 - 3 - Pizza calabresa: Pizza de frango com catupiry, borda recheadaA green button at the bottom right says "Concluir pedido".



Search.. Ctrl + P

Invite

Pizzaria



Sujeito Programador ▾

Run

GET http://localhost:3333/order/detail

Send ▾

200 OK

15.1 ms

1512 B

7 Days Ago ▾

Base Environment

Add Cookies

Add Certificates

Filter

order

PUT Finalizar Pedido (Finish Order)

GET Detalhes de um Pedido (Detail Order)

GET Listar Pedidos (List Orders)

PUT Enviar Pedido (Send Order)

POST Adicionar um Item ao Pedido

DEL Remover Item

POST Criar uma ORDER - Abrir Mesa

DEL Deletar uma ORDER - Pedido

product

category

users

master



Params 1

Body

Auth

Headers 3

Scripts

Docs

Preview

Headers 8

Cookies

Tests 0 / 0

→ Mock

Console

URL PREVIEW

http://localhost:3333/order/detail?order_id=0d0fef8f-9947-4e0a-94d2-dc2722b9cee0

QUERY PARAMETERS

Import from URL Bulk Edit

+ Add

Delete all

Description

order_id

0d0fef8f-9947-4e0a-94d2-dc27...

PATH PARAMETERS

Path parameters are url path segments that start with a colon ':' e.g. ':id'

Preview ▾

```

1 [ 
2 {
3   "id": "1fc66991-1e7e-42a7-a75e-217ccb149c4d",
4   "amount": 2,
5   "created_at": "2024-11-26T14:14:15.760Z",
6   "update_at": "2024-11-26T14:14:15.760Z",
7   "order_id": "0d0fef8f-9947-4e0a-94d2-dc2722b9cee0",
8   "product_id": "ca5c64fa-e4dd-4b67-9c33-2c4961d4b0b1",
9   "product": {
10     "id": "ca5c64fa-e4dd-4b67-9c33-2c4961d4b0b1",
11     "name": "Coca lata",
12     "price": "5",
13     "description": "Coca-Cola lata 350ml",
14     "banner": "765f54ffa7f05a9ff902a1a4b85f0e3b-coca-cola lata
350ml.jpg",
15     "created_at": "2024-11-25T22:17:10.368Z",
16     "update_at": "2024-11-25T22:17:10.368Z",
17     "category_id": "00d8f79f-5536-429a-bd59-7a82f18700ac"
18   },
19   "order": {
20     "id": "0d0fef8f-9947-4e0a-94d2-dc2722b9cee0",
21     "table": 55,
22     "status": false,
23     "draft": false,
24     "name": null,
25     "created_at": "2024-11-26T01:30:26.021Z",
26     "update_at": "2024-11-26T01:30:26.021Z"
27   }
28 },

```

\$.store.books[*].author

Preferences

Online

Made with ❤ by Kong

File Edit View Ultimate Tools Help localhost:5432/pizzaria - Beekeeper Studio

pizzaria C + Query #1 orders [all] +

Filter ENTITIES 217 +

public

- _prisma_migrations
- categories
- items
- orders
- products
- users

information_schema

pg_catalog

id equals Enter Value

| | id | table | status | draft | name | created_at... | update_at |
|---|---------------------------------------|-------|--------|-------|--------|----------------|--------------------------|
| 1 | 0d0fef8f-9947-4e0a-94d2-dc2722b9cee0 | 55 | false | false | (NULL) | 2024-11-26 ... | 2024-11-26 01:30:26.0... |
| 2 | c9019c83-9a02-4152-a475-21f07e1e39... | 9 | true | false | (NULL) | 2024-11-26 ... | 2024-11-26 15:36:31.8... |
| 3 | ee42b24f-cdc5-48fd-87d3-1758a06afae3 | 35 | false | false | Jorge | 2024-11-29 ... | 2024-11-29 02:28:31.9... |

File Edit View Ultimate Tools Help localhost:5432/pizzaria - Beekeeper Studio

pizzaria C + Query #1 orders [all] +

Filter ENTITIES 217 +

public

- _prisma_migrations
- categories
- items
- orders
- products
- users

information_schema

pg_catalog

id equals Enter Value

| | id | table | status | draft | name | created_at... | update_at |
|---|---------------------------------------|-------|--------|-------|--------|----------------|--------------------------|
| 1 | 0d0fef8f-9947-4e0a-94d2-dc2722b9cee0 | 55 | false | false | (NULL) | 2024-11-26 ... | 2024-11-26 01:30:26.0... |
| 2 | c9019c83-9a02-4152-a475-21f07e1e39... | 9 | true | false | (NULL) | 2024-11-26 ... | 2024-11-26 15:36:31.8... |
| 3 | ee42b24f-cdc5-48fd-87d3-1758a06afae3 | 35 | false | false | Jorge | 2024-11-29 ... | 2024-11-29 02:28:31.9... |

Edit in modal Shift+Enter

Set as NULL

Copy Ctrl+C

Copy as TSV for Excel

Copy as JSON

Copy as Markdown

Copy as SQL

Paste Ctrl+V

Clone row Ctrl+D

Delete row Delete



Search.. Ctrl + P

Invite

Pizzaria



Sujeito Programador ▾

Run

GET http://localhost:3333/order/detail

Send ▾

200 OK

131 ms

765 B

Just Now ▾

Base Environment

Add Cookies

Add Certificates

Filter

order

PUT Finalizar Pedido (Finish Order)

GET Detalhes de um Pedido (Detail Order)

GET Listar Pedidos (List Orders)

PUT Enviar Pedido (Send Order)

POST Adicionar um Item ao Pedido

DEL Remover Item

POST Criar uma ORDER - Abrir Mesa

DEL Deletar uma ORDER - Pedido

product

category

users

master



URL PREVIEW

http://localhost:3333/order/detail?order_id=ee42b24f-cdc5-48fd-87d3-1758a06afae3

QUERY PARAMETERS

Import from URL Bulk Edit

+ Add Delete all Description

order_id

ee42b24f-cdc5-48fd-87d3-1758

PATH PARAMETERS

Path parameters are url path segments that start with a colon ':' e.g. ':id'

Preview ▾

```

1 [ 
2 {
3   "id": "1900a07a-65b4-409d-812b-b8aa9a5c7ace",
4   "amount": 2,
5   "created_at": "2024-11-29T02:41:38.002Z",
6   "update_at": "2024-11-29T02:41:38.002Z",
7   "order_id": "ee42b24f-cdc5-48fd-87d3-1758a06afae3",
8   "product_id": "6ad3ded6-115b-468c-89eb-25de1a945944",
9   "product": {
10     "id": "6ad3ded6-115b-468c-89eb-25de1a945944",
11     "name": "Burger Triplo Dev",
12     "price": "50",
13     "description": "Hamburger Triplo com Cheddar",
14     "banner": "9b5565412cabf8912fd21c9c26cfaca6-hamburger
triplo.jpg",
15     "created_at": "2024-11-29T02:14:03.556Z",
16     "update_at": "2024-11-29T02:14:03.556Z",
17     "category_id": "1e4ffc20-04b2-46db-96f0-ceea3ca2975a"
18   },
19   "order": {
20     "id": "ee42b24f-cdc5-48fd-87d3-1758a06afae3",
21     "table": 35,
22     "status": false,
23     "draft": false,
24     "name": "Jorge",
25     "created_at": "2024-11-29T02:28:31.944Z",
26     "update_at": "2024-11-29T02:28:31.944Z"
27   }
28 }

```

\$.store.books[*].author

Preferences

Online

Made with ❤ by Kong



Search.. Ctrl + P

Invite

Pizzaria



Sujeito Programador ▾

Run

GET ▾ http://localhost:3333/order/detail

Send ▾

200 OK

131 ms

765 B

Just Now ▾

Base Environment

Add Cookies

Add Certificates

Filter

order

PUT Finalizar Pedido (Finish Order)

GET Detalhes de um Pedido (Deta...)

GET Listar Pedidos (List Orders)

PUT Enviar Pedido (Send Order)

POST Adicionar um Item ao Pedido

DEL Remover Item

POST Criar uma ORDER - Abrir Mesa

DEL Deletar uma ORDER - Pedido

product

category

users

master



Params 1

Body

Auth 1

Headers 3

Scripts

Docs

Preview

Headers 8

Cookies

Tests 0 / 0

→ Mock

Console

URL PREVIEW

http://localhost:3333/order/detail?order_id=ee42b24f-cdc5-48fd-87d3-1758a06afae3

QUERY PARAMETERS

Import from URL Bulk Edit

+ Add Delete all Description

order_id

ee42b24f-cdc5-48fd-87d3-1758

PATH PARAMETERS

Path parameters are url path segments that start with a colon ':' e.g. ':id'

Preview ▾

```

1  [
2  {
3      "id": "1900a07a-65b4-409d-812b-b8aa9a5c7ace",
4      "amount": 2,
5      "created_at": "2024-11-29T02:41:38.002Z",
6      "update_at": "2024-11-29T02:41:38.002Z",
7      "order_id": "ee42b24f-cdc5-48fd-87d3-1758a06afae3",
8      "product_id": "6ad3ded6-115b-468c-89eb-25de1a945944",
9      "product": {
10          "id": "6ad3ded6-115b-468c-89eb-25de1a945944",
11          "name": "Burger Triplo Dev",
12          "price": "50",
13          "description": "Hamburger Triplo com Cheddar",
14          "banner": "9b5565412cabf8912fd21c9c26cfaca6-hamburger
triplo.jpg",
15          "created_at": "2024-11-29T02:14:03.556Z",
16          "update_at": "2024-11-29T02:14:03.556Z",
17          "category_id": "1e4ffc20-04b2-46db-96f0-ceea3ca2975a"
18      },
19      "order": {
20          "id": "ee42b24f-cdc5-48fd-87d3-1758a06afae3"
21          "table": 35,
22          "status": false,
23          "draft": false,
24          "name": "Jorge",
25          "created_at": "2024-11-29T02:28:31.944Z",
26          "update_at": "2024-11-29T02:28:31.944Z"
27      }
28  }

```

Search with Google

Copy

Select All

\$.store.books[*].author

Online

Made with ❤ by Kong



Search.. Ctrl + P

Invite

Pizzaria ▾



Sujeito Programador ▾

Run

POST ▾ http://localhost:3333/order/add

Send ▾

200 OK

14.5 ms

237 B

5 Days Ago ▾

Params

Body

Auth

Headers

Scripts

Docs

Preview

Headers

Cookies

Tests 0 / 0

→ Mock

Console

JSON ▾

```
1 ▶ {  
2   "order_id": "ee42b24f-cdc5-48fd-87d3-1758a06afae3",  
3   "product_id": "6ad3ded6-115b-468c-89eb-25de1a945944",  
4   "amount": 2  
5 }
```

Preview ▾

```
1 ▶ {  
2   "id": "1900a07a-65b4-409d-812b-b8aa9a5c7ace",  
3   "amount": 2,  
4   "created_at": "2024-11-29T02:41:38.002Z",  
5   "update_at": "2024-11-29T02:41:38.002Z",  
6   "order_id": "ee42b24f-cdc5-48fd-87d3-1758a06afae3",  
7   "product_id": "6ad3ded6-115b-468c-89eb-25de1a945944"  
8 }
```

Base Environment

Add Cookies

Add Certificates

Filter

order

PUT Finalizar Pedido (Finish Order)

GET Detalhes de um Pedido (Deta...)

GET Listar Pedidos (List Orders)

PUT Enviar Pedido (Send Order)

POST Adicionar um Item ao Pedido ▾

DEL Remover Item

POST Criar uma ORDER - Abrir Mesa

DEL Deletar uma ORDER - Pedido

product

category

users

master



Beautify JSON

\$.store.books[*].author



Preferences

Online

Made with ❤ by Kong



Search.. Ctrl + P

Invite

Pizzaria



Sujeito Programador ▾

Run

GET http://localhost:3333/category/product

Send ▾

200 OK

93.8 ms

987 B

5 Days Ago ▾

Base Environment

Add Cookies

Add Certificates

Filter

POST Adicionar um Item ao Pedido

DEL Remover Item

POST Criar uma ORDER - Abrir Mesa

DEL Deletar uma ORDER - Pedido

product

POST Cadastrar Produto

GET Listar Produto por Categoria ▾

category

users

POST Criar usuário

POST Logar usuário

GET Detalhes do usuário

master

URL PREVIEW
http://localhost:3333/category/product?category_id=1e4ffc20-04b2-46db-96f0-ceea3ca2975a

QUERY PARAMETERS

Import from URL Bulk Edit

+ Add Delete all Description

category_id

1e4ffc20-04b2-46db-96f0-ceea3ca2975a

PATH PARAMETERS

Path parameters are url path segments that start with a colon ':' e.g. ':id'

Preview

Headers ▾

Cookies

Tests 0 / 0

→ Mock

Console

Preview ▾

```

1 [
2 {
3   "id": "96eb4477-824b-4137-8251-dbce1d72f755",
4   "name": "Dev Burger Salada",
5   "price": "40",
6   "description": "Burgão Dev de Salada com 1 hambúrguer de 160g",
7   "banner": "c83f6cc7a8b3dea100c14366e0808e99-hamburger.jpg",
8   "created_at": "2024-11-29T01:06:24.050Z",
9   "update_at": "2024-11-29T01:06:24.050Z",
10  "category_id": "1e4ffc20-04b2-46db-96f0-ceea3ca2975a"
11 },
12 {
13   "id": "0cd97aa3-1583-458e-b0ef-549fd1f40e37",
14   "name": "Burger Completo Duplo",
15   "price": "45",
16   "description": "Hambúrguer Duplo Top",
17   "banner": "040cfab507866ce68a0ce7fdc60e8cc8-hamburger duplo.jpg",
18   "created_at": "2024-11-29T01:57:17.362Z",
19   "update_at": "2024-11-29T01:57:17.362Z",
20   "category_id": "1e4ffc20-04b2-46db-96f0-ceea3ca2975a"
21 },
22 {
23   "id": "6ad3ded6-115b-468c-89eb-25de1a945944",
24   "name": "Burger Triplo Dev",
25   "price": "50",
26   "description": "Hambúrguer Triplo com Cheddar",
27   "banner": "9b5565412cabf8912fd21c9c26cfaca6-hamburger triplo.jpg",
28   "created_at": "2024-11-29T02:14:03.556Z",
29   "update_at": "2024-11-29T02:14:03.556Z",

```

\$store.books[*].author

Online

Made with ❤ by Kong



Preferences



Search.. Ctrl + P

Invite

Pizzaria



Sujeito Programador ▾

Run

POST ▾ http://localhost:3333/order/add

Send ▾

200 OK

155 ms

237 B

Just Now ▾

Base Environment

Add Cookies

Add Certificates

Filter

order

PUT Finalizar Pedido (Finish Order)

GET Detalhes de um Pedido (Deta...)

GET Listar Pedidos (List Orders)

PUT Enviar Pedido (Send Order)

POST Adicionar um Item ao Pedido ▾

DEL Remover Item

POST Criar uma ORDER - Abrir Mesa

DEL Deletar uma ORDER - Pedido

product

POST Cadastrar Produto

GET Listar Produto por Categoria

master



Beautify JSON

Preview

Headers 8

Cookies

Tests 0 / 0

→ Mock

Console

Preview ▾

```
1 {  
2   "order_id": "ee42b24f-cdc5-48fd-87d3-1758a06afae3",  
3   "product_id": "96eb4477-824b-4137-8251-dbce1d72f755",  
4   "amount": 1  
5 }
```

```
1 {  
2   "id": "471a49ae-7db2-4dd8-8f6d-476040a6e166",  
3   "amount": 1,  
4   "created_at": "2024-12-03T23:07:45.492Z",  
5   "update_at": "2024-12-03T23:07:45.492Z",  
6   "order_id": "ee42b24f-cdc5-48fd-87d3-1758a06afae3",  
7   "product_id": "96eb4477-824b-4137-8251-dbce1d72f755"  
8 }
```

\$.store.books[*].author



Preferences

Online

Made with ❤ by Kong



Search.. Ctrl + P

Invite

Pizzaria



Sujeito Programador ▾

Run

GET ▾ http://localhost:3333/order/detail

Send ▾

200 OK

66 ms

1538 B

Just Now ▾

Base Environment

Add Cookies

Add Certificates

Filter

order

PUT Finalizar Pedido (Finish Order)

GET Detalhes de um Pedido (Deta... ▾

GET Listar Pedidos (List Orders)

PUT Enviar Pedido (Send Order)

POST Adicionar um Item ao Pedido

DEL Remover Item

POST Criar uma ORDER - Abrir Mesa

DEL Deletar uma ORDER - Pedido

product

POST Cadastrar Produto

GET Listar Produto por Categoria

master



Preferences

Params 1

Body

Auth

Headers 3

Scripts

Docs

Preview

Headers 8

Cookies

Tests 0 / 0

→ Mock

Console

URL PREVIEW

http://localhost:3333/order/detail?order_id=ee42b24f-cdc5-48fd-87d3-1758a06afae3

QUERY PARAMETERS

Import from URL Bulk Edit

+ Add

Delete all

Description

order_id

ee42b24f-cdc5-48fd-87d3-1758

PATH PARAMETERS

Path parameters are url path segments that start with a colon ':' e.g. ':id'

Preview ▾

```

1  [
2   {
3     "id": "1900a07a-65b4-409d-812b-b8aa9a5c7ace",
4     "amount": 2,
5     "created_at": "2024-11-29T02:41:38.002Z",
6     "update_at": "2024-11-29T02:41:38.002Z",
7     "order_id": "ee42b24f-cdc5-48fd-87d3-1758a06afae3",
8     "product_id": "6ad3ded6-115b-468c-89eb-25de1a945944",
9     "product": {
10       "id": "6ad3ded6-115b-468c-89eb-25de1a945944",
11       "name": "Burger Triplo Dev",
12       "price": "50",
13       "description": "Hamburger Triplo com Cheddar",
14       "banner": "9b5565412cabf8912fd21c9c26cfaca6-hamburger
triplo.jpg",
15       "created_at": "2024-11-29T02:14:03.556Z",
16       "update_at": "2024-11-29T02:14:03.556Z",
17       "category_id": "1e4ffc20-04b2-46db-96f0-ceea3ca2975a"
18     },
19     "order": {
20       "id": "ee42b24f-cdc5-48fd-87d3-1758a06afae3",
21       "table": 35,
22       "status": false,
23       "draft": false,
24       "name": "Jorge",
25       "created_at": "2024-11-29T02:28:31.944Z",
26       "update_at": "2024-11-29T02:28:31.944Z"
27     }
28   },

```

\$.store.books[*].author

Online

Made with ❤ by Kong



Search.. Ctrl + P

Invite

Pizzaria

Sujeito Programador ▾

Run

GET http://localhost:3333/order/detail

Send

200 OK

66 ms

1538 B

1 Minute Ago ▾



Base Environment

Add Cookies

Add Certificates

Filter

order

PUT Finalizar Pedido (Finish Order)

GET Detalhes de um Pedido (Deta...)

GET Listar Pedidos (List Orders)

PUT Enviar Pedido (Send Order)

POST Adicionar um Item ao Pedido

DEL Remover Item

POST Criar uma ORDER - Abrir Mesa

DEL Deletar uma ORDER - Pedido

product

POST Cadastrar Produto

GET Listar Produto por Categoria

master



Preferences

Params 1

Body

Auth

Headers 3

Scripts

Docs

Preview

Headers 8

Cookies

Tests 0 / 0

→ Mock

Console

URL PREVIEW

http://localhost:3333/order/detail?order_id=ee42b24f-cdc5-48fd-87d3-1758a06afae3

QUERY PARAMETERS

Import from URL Bulk Edit

+ Add

Delete all

Description

order_id

ee42b24f-cdc5-48fd-87d3-1758

PATH PARAMETERS

Path parameters are url path segments that start with a colon ':' e.g. ':id'

Preview

```

27
28 },
29 {
30   "id": "471a49ae-7db2-4dd8-8f6d-476040a6e166",
31   "amount": 1,
32   "created_at": "2024-12-03T23:07:45.492Z",
33   "update_at": "2024-12-03T23:07:45.492Z",
34   "order_id": "ee42b24f-cdc5-48fd-87d3-1758a06afae3",
35   "product_id": "96eb4477-824b-4137-8251-dbce1d72f755",
36   "product": {
37     "id": "96eb4477-824b-4137-8251-dbce1d72f755",
38     "name": "Dev Burger Salada",
39     "price": "40",
40     "description": "Burgão Dev de Salada com 1 hambúrguer de 160g",
41     "banner": "c83f6cc7a8b3dea100c14366e0808e99-hamburger.jpg",
42     "created_at": "2024-11-29T01:06:24.050Z",
43     "update_at": "2024-11-29T01:06:24.050Z",
44     "category_id": "1e4ffc20-04b2-46db-96f0-ceea3ca2975a"
45   },
46   "order": {
47     "id": "ee42b24f-cdc5-48fd-87d3-1758a06afae3",
48     "table": 35,
49     "status": false,
50     "draft": false,
51     "name": "Jorge",
52     "created_at": "2024-11-29T02:28:31.944Z",
53     "update_at": "2024-11-29T02:28:31.944Z"
54   }
55 }
56 ]

```

\$.store.books[*].author

Online

Made with ❤ by Kong



Search.. Ctrl + P

Invite

Pizzaria ▾



Sujeito Programador ▾

Run

GET ▾ http://localhost:3333/order/detail

Send ▾

200 OK

66 ms

1538 B

2 Minutes Ago ▾

Base Environment

Add Cookies

Add Certificates

Filter

order

PUT Finalizar Pedido (Finish Order)

GET Detalhes de um Pedido (Deta... ▾

GET Listar Pedidos (List Orders)

PUT Enviar Pedido (Send Order)

POST Adicionar um Item ao Pedido

DEL Remover Item

POST Criar uma ORDER - Abrir Mesa

DEL Deletar uma ORDER - Pedido

product

POST Cadastrar Produto

GET Listar Produto por Categoria

master



Preferences

Params 1

Body

Auth

Headers 3

Scripts

Docs

Preview

Headers 8

Cookies

Tests 0 / 0

→ Mock

Console

URL PREVIEW

http://localhost:3333/order/detail?order_id=ee42b24f-cdc5-48fd-87d3-1758a06afae3



QUERY PARAMETERS

Import from URL Bulk Edit

+ Add

Delete all

Description

order_id

ee42b24f-cdc5-48fd-87d3-1758



PATH PARAMETERS

Path parameters are url path segments that start with a colon ':' e.g. ':id'



```
1 [  
2 { ↪ 8 ↪ },  
29 { ↪ 8 ↪ }  
56 ]
```

\$.store.books[*].author

Online

Made with ❤ by Kong



Search.. Ctrl + P

Invite

Pizzaria



Sujeito Programador ▾

Run

GET http://localhost:3333/order/detail

Send

200 OK

66 ms

1538 B

3 Minutes Ago ▾

Base Environment

Add Cookies

Add Certificates

Filter

order

PUT Finalizar Pedido (Finish Order)

GET Detalhes de um Pedido (Deta...)

GET Listar Pedidos (List Orders)

PUT Enviar Pedido (Send Order)

POST Adicionar um Item ao Pedido

DEL Remover Item

POST Criar uma ORDER - Abrir Mesa

DEL Deletar uma ORDER - Pedido

product

POST Cadastrar Produto

GET Listar Produto por Categoria

master



Params 1

Body

Auth

Headers 3

Scripts

Docs

Preview

Headers 8

Cookies

Tests 0 / 0

→ Mock

Console

URL PREVIEW

http://localhost:3333/order/detail?order_id=ee42b24f-cdc5-48fd-87d3-1758a06afae3

QUERY PARAMETERS

Import from URL Bulk Edit

+ Add

Delete all

Description

order_id

ee42b24f-cdc5-48fd-87d3-1758

PATH PARAMETERS

Path parameters are url path segments that start with a colon ':' e.g. ':id'

Preview ▾

```

1 [ 
2 {
3   "id": "1900a07a-65b4-409d-812b-b8aa9a5c7ace",
4   "amount": 2,
5   "created_at": "2024-11-29T02:41:38.002Z",
6   "update_at": "2024-11-29T02:41:38.002Z",
7   "order_id": "ee42b24f-cdc5-48fd-87d3-1758a06afae3",
8   "product_id": "6ad3ded6-115b-468c-89eb-25de1a945944",
9   "product": {
10     "id": "6ad3ded6-115b-468c-89eb-25de1a945944",
11     "name": "Burger Triplo Dev",
12     "price": "50",
13     "description": "Hamburger Triplo com Cheddar",
14     "banner": "9b5565412cabf8912fd21c9c26cfaca6-hamburger
triplo.jpg",
15     "created_at": "2024-11-29T02:14:03.556Z",
16     "update_at": "2024-11-29T02:14:03.556Z",
17     "category_id": "1e4ffc20-04b2-46db-96f0-ceea3ca2975a"
18   },
19   "order": {
20     "id": "ee42b24f-cdc5-48fd-87d3-1758a06afae3",
21     "table": 35,
22     "status": false,
23     "draft": false,
24     "name": "Jorge",
25     "created_at": "2024-11-29T02:28:31.944Z",
26     "update_at": "2024-11-29T02:28:31.944Z"
27   }
28 },

```

\$.store.books[*].author

Preferences

Online

Made with ❤ by Kong



Search.. Ctrl + P

Invite

Pizzaria ▾



Sujeito Programador ▾

Run

GET ▾ http://localhost:3333/order/detail

Send ▾

200 OK

66 ms

1538 B

3 Minutes Ago ▾

Base Environment

Add Cookies

Add Certificates

Filter

order

PUT Finalizar Pedido (Finish Order)

GET Detalhes de um Pedido (Deta... ▾

GET Listar Pedidos (List Orders)

PUT Enviar Pedido (Send Order)

POST Adicionar um Item ao Pedido

DEL Remover Item

POST Criar uma ORDER - Abrir Mesa

DEL Deletar uma ORDER - Pedido

product

POST Cadastrar Produto

GET Listar Produto por Categoria

master



Preferences

Params 1

Body

Auth

Headers 3

Scripts

Docs

Preview

Headers 8

Cookies

Tests 0 / 0

→ Mock

Console

URL PREVIEW

http://localhost:3333/order/detail?order_id=ee42b24f-cdc5-48fd-87d3-1758a06afae3

QUERY PARAMETERS

Import from URL Bulk Edit

+ Add

Delete all

Description

order_id

ee42b24f-cdc5-48fd-87d3-1758

PATH PARAMETERS

Path parameters are url path segments that start with a colon ':' e.g. ':id'

Preview ▾

```

27   }
28 },
29 {
30   "id": "471a49ae-7db2-4dd8-8f6d-476040a6e166",
31   "amount": 1,
32   "created_at": "2024-12-03T23:07:45.492Z",
33   "update_at": "2024-12-03T23:07:45.492Z",
34   "order_id": "ee42b24f-cdc5-48fd-87d3-1758a06afae3",
35   "product_id": "96eb4477-824b-4137-8251-dbce1d72f755",
36   "product": {
37     "id": "96eb4477-824b-4137-8251-dbce1d72f755",
38     "name": "Dev Burger Salada",
39     "price": "40",
40     "description": "Burgão Dev de Salada com 1 hambúrguer de 160g",
41     "banner": "c83f6cc7a8b3dea100c14366e0808e99-hamburger.jpg",
42     "created_at": "2024-11-29T01:06:24.050Z",
43     "update_at": "2024-11-29T01:06:24.050Z",
44     "category_id": "1e4ffc20-04b2-46db-96f0-ceea3ca2975a"
45   },
46   "order": {
47     "id": "ee42b24f-cdc5-48fd-87d3-1758a06afae3",
48     "table": 35,
49     "status": false,
50     "draft": false,
51     "name": "Jorge",
52     "created_at": "2024-11-29T02:28:31.944Z",
53     "update_at": "2024-11-29T02:28:31.944Z"
54   }
55 }
```

\$.store.books[*].author

Online

Made with ❤ by Kong



Search.. Ctrl + P

Invite

Pizzaria



Sujeito Programador ▾

Run

GET http://localhost:3333/order/detail

Send ▾

200 OK

66 ms

1538 B

5 Minutes Ago ▾

Base Environment

Add Cookies

Add Certificates

Filter

order

PUT Finalizar Pedido (Finish Order)

GET Detalhes de um Pedido (Deta... ▾

GET Listar Pedidos (List Orders)

PUT Enviar Pedido (Send Order)

POST Adicionar um Item ao Pedido

DEL Remover Item

POST Criar uma ORDER - Abrir Mesa

DEL Deletar uma ORDER - Pedido

product

POST Cadastrar Produto

GET Listar Produto por Categoria

master



Preview

Headers 8

Cookies

Tests 0 / 0

→ Mock Console

Preview ▾

```
1 [
2 {
3   "id": 8
4 },
5 {
6   "id": 8
7 }
8 ]
```

URL PREVIEW

http://localhost:3333/order/detail?order_id=ee42b24f-cdc5-48fd-87d3-1758a06afae3

QUERY PARAMETERS

Import from URL Bulk Edit

+ Add

Delete all

Description

order_id

ee42b24f-cdc5-48fd-87d3-1758

▼

✓

Delete

PATH PARAMETERS

Path parameters are url path segments that start with a colon ':' e.g. ':id'

X

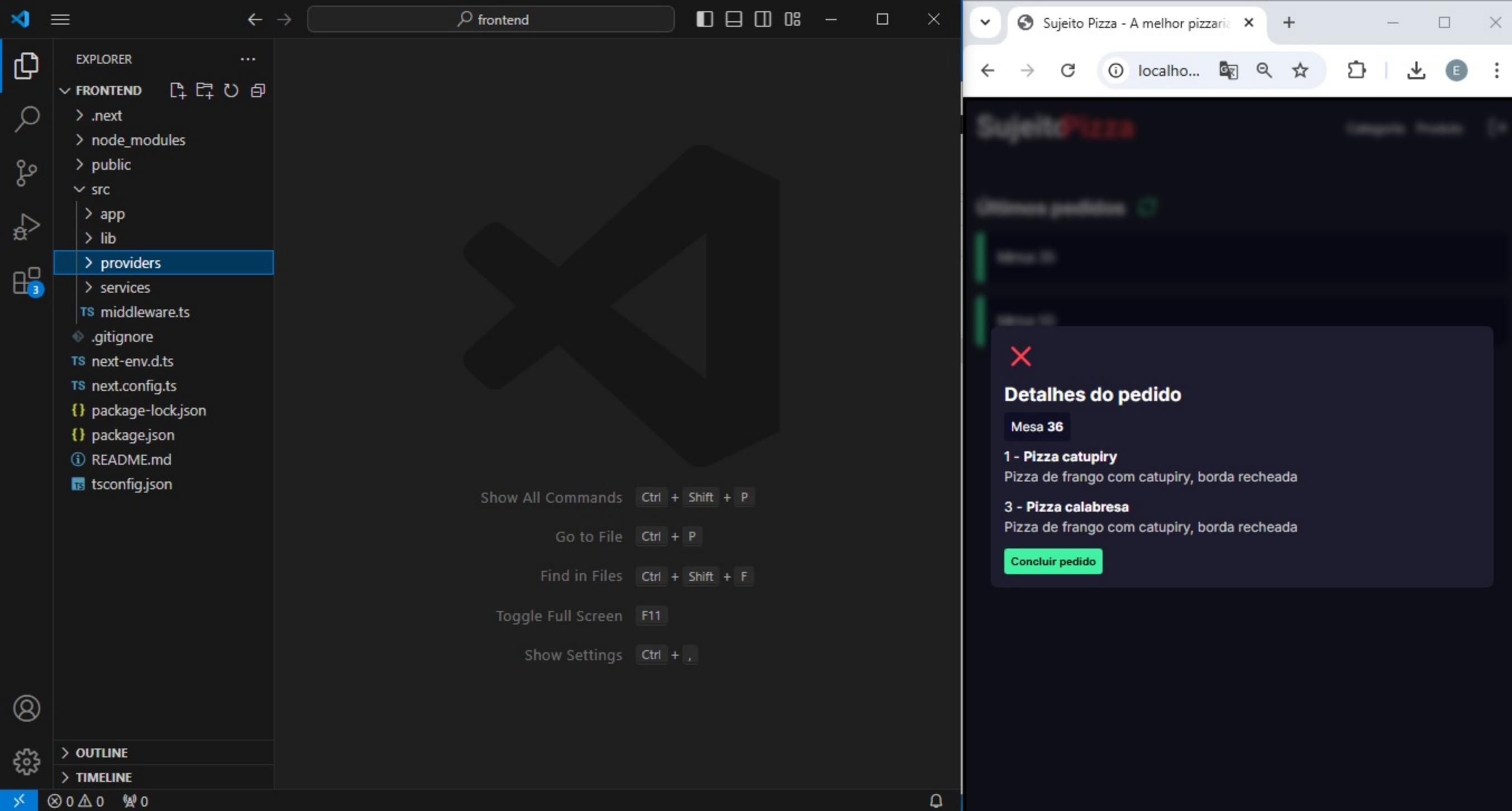
\$.store.books[*].author



Preferences

Online

Made with ❤ by Kong



The image shows a developer's workspace with two main windows: a code editor in the foreground and a browser preview in the background.

Code Editor (VS Code):

- File Structure:** The left sidebar shows a project structure under "FRONTEND".
 - src
 - .next
 - node_modules
 - public
 - src
 - app
 - lib
 - providers
 - order.tsx
 - services
 - middleware.ts
 - .gitignore
 - next-env.d.ts
 - next.config.ts
 - package-lock.json
 - package.json
 - README.md
 - tsconfig.json

order.tsx Content:

```
use client
import { createContext, ReactNode, useState } from 'react'

type OrderContextData = {
  isOpen: boolean;
  onRequestOpen: () => void;
  onRequestClose: () => void;
}

type OrderProviderProps = {
  children: ReactNode;
}

export const OrderContext = createContext({} as OrderContextData)

export function OrderProvider({ children }: OrderProviderProps) {
  const [isOpen, setIsOpen] = useState(false);

  function onRequestOpen() {
    setIsOpen(true);
  }

  function onRequestClose() {
    setIsOpen(false);
  }

  return (
    <OrderContext.Provider>
      <value={{
        isOpen,
        onRequestOpen,
        onRequestClose
      }}>
        {children}
      </value>
    </OrderContext.Provider>
  );
}
```

Browser Preview:

The browser window displays a dark-themed application interface titled "Sujeito Pizza - A melhor pizzaria". The main content area shows a list of pizzas and a modal dialog.

Modal Dialog (Detalhes do pedido):

- Mesa 36:**
 - 1 - Pizza catupiry**
Pizza de frango com catupiry, borda recheada
 - 3 - Pizza calabresa**
Pizza de frango com catupiry, borda recheada
- Concluir pedido** button

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- order.tsx:** The active file contains the following TypeScript code:

```
src > providers > order.tsx > OrderProvider
33     }})
34     ...
35     {children}
36   </OrderContext.Provider>
37 )
38 }
39
40
41
```

Browser Preview:

The browser window displays a dark-themed interface for "Sujeito Pizza - A melhor pizzaria". The page shows a header with the logo and some blurred text. Below it, there's a sidebar with a red "X" icon and the text "Detalhes do pedido". The main content area lists two items:

- Mesa 36**
 - 1 - Pizza catupiry**
Pizza de frango com catupiry, borda recheada
 - 3 - Pizza calabresa**
Pizza de frango com catupiry, borda recheada

A green "Concluir pedido" button is at the bottom right of the content area.

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "layout.tsx" is the active file.
- Content:** The code defines a `DashboardLayout` component that wraps children with a `Header` and an `OrderProvider`.

```
src > app > dashboard > layout.tsx > DashboardLayout
1 import { Header } from "./components/header"
2 import { OrderProvider } from "@/providers/order"
3
4 export default function DashboardLayout({ children }: {
5   children: React.ReactNode
6 }) {
7   return(
8     <>
9       <Header/>
10      <OrderProvider>
11        <children>
12        </OrderProvider>
13      </>
14    )
15 }
```

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localhost:3001
- Content:** A dark-themed pizza ordering interface. A modal dialog is open, titled "Detalhes do pedido", showing an order for "Mesa 36". It lists two items:
 - 1 - Pizza catupiry**: Pizza de frango com catupiry, borda recheada
 - 3 - Pizza calabresa**: Pizza de frango com catupiry, borda recheadaA green "Concluir pedido" button is at the bottom right of the modal.

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file.
- Content:** The code is a React component named "Orders". It imports "use client", "react", "styles", "RefreshCw", "OrderProps", "Modalorder", and "OrderContext". It defines a props interface "Props" with an "orders" array. The component returns a main container with a header section for the last orders and a list of orders.
- Status Bar:** Shows 4 selections (138 characters selected), Spaces: 2, CRLF, TypeScript JSX, Prettier, and a bell icon.

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localho... (localhost)
- Content:** A dark-themed web page titled "Sujeito Pizza". It displays a header with the logo and navigation links. Below it, there's a section for the last orders and a modal window titled "Detalhes do pedido" for "Mesa 36". The modal shows two pizzas: "1 - Pizza catupiry" and "3 - Pizza calabresa", both described as "Pizza de frango com catupiry, borda recheada". A green button at the bottom right says "Concluir pedido".

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file.
- Content:** The code is a TypeScript component for displaying orders. It includes an H1 title, a refresh button, and a list of orders. Each order item has a button labeled with its table number (e.g., Mesa 36). A modal component is also imported.

```
src > app > dashboard > components > orders > index.tsx > Orders
22    <h1>Últimos pedidos</h1>
23    <button>
24      | <RefreshCw size={24} color="#3ffffa3" />
25    </button>
26  </section>
27
28  <section className={styles.listOrders}>
29    {orders.map( order => (
30      <button
31        key={order.id}
32        className={styles.orderItem}
33        >
34          <div className={styles.tag}></div>
35          <span>Mesa {order.table}</span>
36        </button>
37    ))}
38  </section>
39  </main>
40
41  <Modalorder/>
42
43  </>
44
45  )
46 }
```

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localho... (localhost)
- Content:** A dark-themed web application interface for managing orders. It shows a header with the Sujeito Pizza logo and a search bar. Below is a list of orders. A modal window is open for "Mesa 36", showing two pizzas: "Pizza catupiry" and "Pizza calabresa".

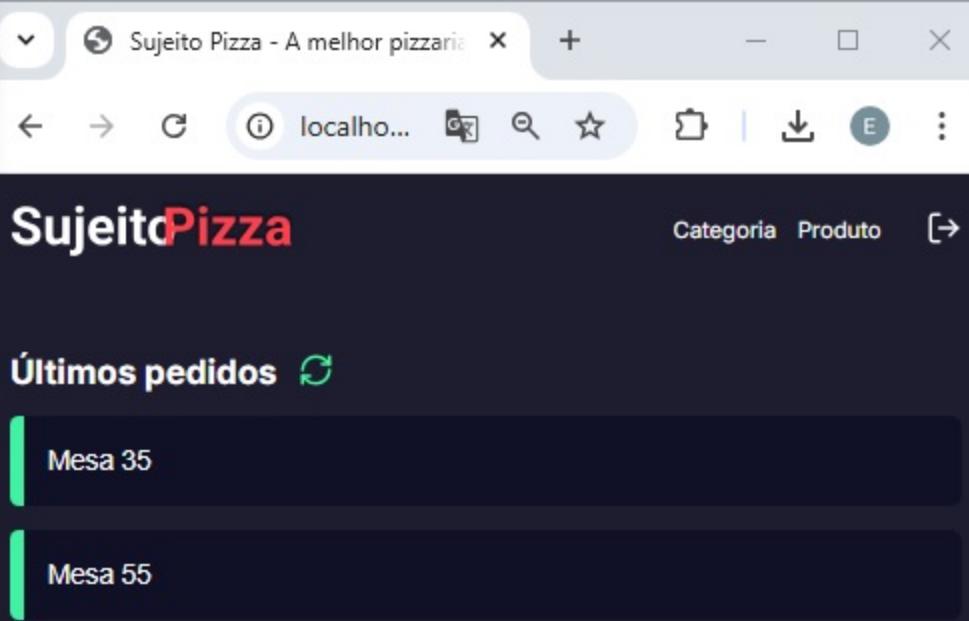
Modal Content (Detalhes do pedido):

- Mesa:** Mesa 36
- Pizza 1:** Pizza catupiry (Pizza de frango com catupiry, borda recheada)
- Pizza 2:** Pizza calabresa (Pizza de frango com catupiry, borda recheada)
- Buttons:** Concluir pedido

The screenshot shows the VS Code interface with the following details:

- Explorer View:** Shows the project structure under "FRONTEND". The "orders" folder is expanded, showing files like "index.tsx", "styles.module.scss", and "page.tsx".
- Search Bar:** Contains the text "frontend".
- Active File:** "index.tsx" is the active file, indicated by the title bar and the highlighted code in the editor.
- Editor Content:** The code displays a component structure for displaying orders. It includes an H1 header, a refresh button, and a list of orders. Each order item contains a button with the order ID and a span showing the table name.
- Status Bar:** Shows "In 43 Col 5 (27 selected)" and "Spaces: 2".

```
src > app > dashboard > components > orders > index.tsx > Orders
22 <h1>Últimos pedidos</h1>
23 <button>
24 | <RefreshCw size={24} color="#3ffffa3" />
25 </button>
26 </section>
27
28 <section className={styles.listOrders}>
29 | {orders.map( order => (
30 |   <button
31 |     key={order.id}
32 |     className={styles.orderItem}
33 |   >
34 |     <div className={styles.tag}></div>
35 |     <span>Mesa {order.table}</span>
36 |   </button>
37 | ))}>
38 </section>
39 </main>
40
41 { isOpen && <Modalorder/> }
42 </>
43 )
44
45 }
46 }
```



The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- order.tsx:** The active file contains TypeScript code for a Context Provider. It defines two types: `OrderContextData` and `OrderProviderProps`. It then creates an `OrderContext` using `createContext` and defines an `OrderProvider` component that uses `useState` to manage the `isOpen` state. It includes functions `onRequestOpen` and `onRequestClose` to update the state.
- index.tsx:** The other file in the tab bar.

Browser Preview:

The browser window displays a pizza delivery application titled "SujeitoPizza". The page shows a header with "SujeitoPizza" and navigation links for "Categoria" and "Produto". Below the header, a section titled "Últimos pedidos" lists recent orders: "Mesa 35" and "Mesa 55".

Bottom status bar: Ln 17, Col 40 (5 selected) Spaces: 4 UTF-8 CRLF TypeScript JSX Prettier

The image shows a developer's environment with two main windows: a code editor and a web browser.

Code Editor (VS Code):

- Explorer:** Shows the project structure under **FRONTEND**, including **src**, **app**, **dashboard**, **product**, **layout.tsx**, **page.tsx**, **fonts**, **signup**, **favicon.ico**, **globals.scss**, **layout.tsx**, **page.module.scss**, and **page.tsx**.
- order.tsx:** The current file being edited. It defines a context provider for managing order requests. The code includes type definitions for **OrderContextData** and **OrderProviderProps**, and implementation for **OrderProvider** which uses **useState** to manage the **isOpen** state and provides functions **onRequestOpen** and **onRequestClose**.

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localho... (localhost)
- Content:** A dark-themed interface for a pizza delivery app. It shows a header with the app logo and some blurred text. Below is a sidebar with icons for **Dashboard**, **Produtos**, **Mesa 36**, and **Logout**. A modal window titled "Detalhes do pedido" is open for "Mesa 36". It lists two pizzas:
 - 1 - Pizza catupiry**: Pizza de frango com catupiry, borda recheada
 - 3 - Pizza calabresa**: Pizza de frango com catupiry, borda recheadaA green button labeled "Concluir pedido" is at the bottom of the modal.

Bottom status bar: Ln 17, Col 40 (4 selected) Spaces: 4 UTF-8 CRLF TypeScript JSX Prettier

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under **FRONTEND**, including **src**, **app**, **dashboard**, **fonts**, **signup**, **favicon.ico**, **globals.scss**, **layout.tsx**, **page.module.scss**, **page.tsx**, **lib**, **providers** (selected), **order.tsx**, **services**, **middleware.ts**, **.gitignore**, **next-env.d.ts**, **next.config.ts**, **package-lock.json**, **package.json**, **README.md**, and **tsconfig.json**.
- order.tsx:** The current file being edited contains TypeScript code for a Context Provider. It defines two types: **OrderContextData** and **OrderProviderProps**. It then creates an **OrderContext** using `createContext` and defines an **OrderProvider** component that uses `useState` to manage the **isOpen** state. It includes functions **onRequestOpen** and **onRequestClose** to update the state.

Browser Preview:

The browser window displays a dark-themed interface for "Sujeito Pizza - A melhor pizzaria". A modal dialog titled "Detalhes do pedido" is open, showing details for "Mesa 36".

- Mesa 36:**
 - 1 - Pizza catupiry:** Pizza de frango com catupiry, borda recheada
 - 3 - Pizza calabresa:** Pizza de frango com catupiry, borda recheada
- A green button labeled "Concluir pedido" is visible at the bottom right of the modal.

At the bottom of the screen, the status bar shows: Ln 17, Col 44 (4 selected) Spaces: 4 UTF-8 CRLF TypeScript JSX Prettier.

The screenshot shows the VS Code interface with the following details:

- Explorer View:** Shows the project structure under "FRONTEND".
- File:** "order.tsx" is open in the editor.
- Content:** The code defines a context provider for managing order requests. It includes type definitions for `OrderContextData` and `OrderProviderProps`, and implementation for `OrderProvider` which uses `useState` to manage the `isOpen` state and provides functions to open and close requests.

```
4  type OrderContextData = {
5    isOpen: boolean;
6    onRequestOpen: () => void;
7    onRequestClose: () => void;
8  }
9
10 type OrderProviderProps = {
11   children: ReactNode;
12 }
13
14 export const OrderContext = createContext({} as OrderContextData)
15
16 export function OrderProvider({ children }: OrderProviderProps){
17   const [isOpen, setIsOpen] = useState(false);
18
19   function onRequestOpen(){
20     setIsOpen(true);
21   }
22
23   function onRequestClose(){
24     setIsOpen(false);
25   }
26
27   return(
28     <OrderContext.Provider
29       value={{
30         isOpen,
31         onRequestOpen,
32         onRequestClose
33       }}
34     >
35       {children}
36     </OrderContext.Provider>
37   )
38 }
```

- Status Bar:** Shows file statistics: Ln 17, Col 40 (5 selected), Spaces: 4, UTF-8, CRLF, TypeScript JSX, Prettier.

The screenshot shows a browser window displaying the "Sujeito Pizza" website. The page features a header with the logo and navigation links for "Categoria" and "Produto". Below the header, there is a section titled "Últimos pedidos" (Recent Orders) with two items listed: "Mesa 35" and "Mesa 55".

UC: Projetos

Assunto:

Projeto Pizzaria – Frontend_02 (NextJS)

Sistema Pizzaria Web

60 – Abrindo Modal

frontend

EXPLORER

FRONTEND

- > .next
- > node_modules
- > public
- src
 - app
 - dashboard
 - > category
 - > components
 - > button
 - > header
 - > modal
 - > orders
 - index.tsx
 - styles.module.scss
 - > product
 - > page.tsx
 - > fonts
 - > signup
 - favicon.ico
 - globals.scss
 - layout.tsx
 - page.module.scss
 - page.tsx
 - > lib
 - providers
 - > OUTLINE
 - > TIMELINE

layout.tsx

```
src > app > dashboard > layout.tsx > DashboardLayout
1 import { Header } from "./components/header"
2 import { OrderProvider } from "@/providers/order"
3
4 export default function DashboardLayout({ children }: {
5   children: React.ReactNode
6 }) {
7   return(
8     <>
9       <Header/>
10      <OrderProvider>
11        <children>
12        </children>
13      </OrderProvider>
14    </>
15  )
}
```

Sujeito Pizza - A melhor pizzaria

localhost

SujeitoPizza

Categoria Produto

Últimos pedidos

- Mesa 35
- Mesa 55

The image shows a developer's workspace with two main windows: a code editor and a browser preview.

Code Editor (Left):

- Explorer:** Shows the project structure. The `order.tsx` file is currently selected in the `src/providers` folder.
- order.tsx Content:** Displays the following TypeScript code:

```
src > providers > order.tsx > OrderProvider
6   onRequestOpen: () => void;
7   onRequestClose: () => void;
8 }
9
10 type OrderProviderProps = {
11   children: ReactNode;
12 }
13
14 export const OrderContext = createContext({} as OrderContextData)
15
16 export function OrderProvider({ children }: OrderProviderProps){
17   const [isOpen, setIsOpen] = useState(false);
18
19   function onRequestOpen(){
20     setIsOpen(true);
21   }
22
23   function onRequestClose(){
24     setIsOpen(false);
25   }
26
27   return(
28     <OrderContext.Provider>
29       value={{
30         isOpen,
31         onRequestOpen,
32         onRequestClose
33       }}
34     >
35       {children}
36     </OrderContext.Provider>
37   )
38 }
```

Browser Preview (Right):

- Title:** Sujeito Pizza - A melhor pizzaria
- Header:** Localhost... (with a refresh icon)
- Content:** The page displays the title "SujeitoPizza" and a section titled "Últimos pedidos" (Last orders) which lists "Mesa 35" and "Mesa 55".
- Navigation:** Includes links for "Categoria" and "Produto" and a search bar.

At the bottom of the screen, there are status indicators: 2 selections (187 characters selected), Spaces: 2, UTF-8, CRLF, TypeScript JSX, Prettier, and a magnifying glass icon.

The image shows a developer's workspace split into two main sections: a code editor on the left and a browser preview on the right.

Code Editor (Left):

- Explorer:** Shows the project structure. The file `order.tsx` is currently selected in the providers folder under `src`.
- Frontend:** Other files visible include `.next`, `node_modules`, `public`, `app`, `lib`, `services`, `middleware.ts`, `.gitignore`, `next-env.d.ts`, `next.config.ts`, `package-lock.json`, `package.json`, `README.md`, and `tsconfig.json`.
- Editor Area:** The code for `order.tsx` is displayed. It defines a context provider (`OrderProvider`) that manages the state of an open request. The code includes type definitions, context creation, and provider implementation.

```
order.tsx
src > providers > order.tsx > OrderProvider > onRequestClose
6   onRequestOpen: () => void;
7   onRequestClose: () => void;
8 }
9
10 type OrderProviderProps = {
11   children: ReactNode;
12 }
13
14 export const OrderContext = createContext({} as OrderContextData)
15
16 export function OrderProvider({ children }: OrderProviderProps){
17   const [isOpen, setIsOpen] = useState(false);
18
19   function onRequestOpen(){
20     setIsOpen(true);
21   }
22
23   function onRequestClose(){
24     setIsOpen(false);
25   }
26
27   return(
28     <OrderContext.Provider
29       value={{
30         isOpen,
31         onRequestOpen,
32         onRequestClose
33       }}
34     >
35       {children}
36     </OrderContext.Provider>
37   )
38 }
```

Browser Preview (Right):

The browser window displays a dark-themed website for "Sujeito Pizza". The title bar says "Sujeito Pizza - A melhor pizzaria". The page content includes a header with "SujeitoPizza" and navigation links for "Categoria" and "Produto". Below the header, a section titled "Últimos pedidos" lists recent orders: "Mesa 35" and "Mesa 55".

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure. The file `order.tsx` is currently selected in the providers folder under `src`.
- Editor:** Displays the content of `order.tsx`. The code defines an `OrderProvider` component that manages the state of a request being open or closed. It uses the `useState` hook to track the `isOpen` state and provides functions to open and close requests.
- Bottom Status Bar:** Shows file statistics (0△0), line/col (Ln 25, Col 4), and encoding (UTF-8 CRLF).

Browser Preview:

- Title Bar:** Shows the title "Sujeito Pizza - A melhor pizzaria" and the URL "localhost:3001".
- Header:** Features the brand name "SujeitoPizza" in large letters, with "Sujeito" in blue and "Pizza" in red. Navigation links include "Categoria" and "Produto".
- Section:** Titled "Últimos pedidos" (Last orders) with a refresh icon.
- Content:** Lists recent orders from "Mesa 35" and "Mesa 55".

Bottom Status Bar: Shows file statistics (0△0), line/col (Ln 25, Col 4), and encoding (UTF-8 CRLF).

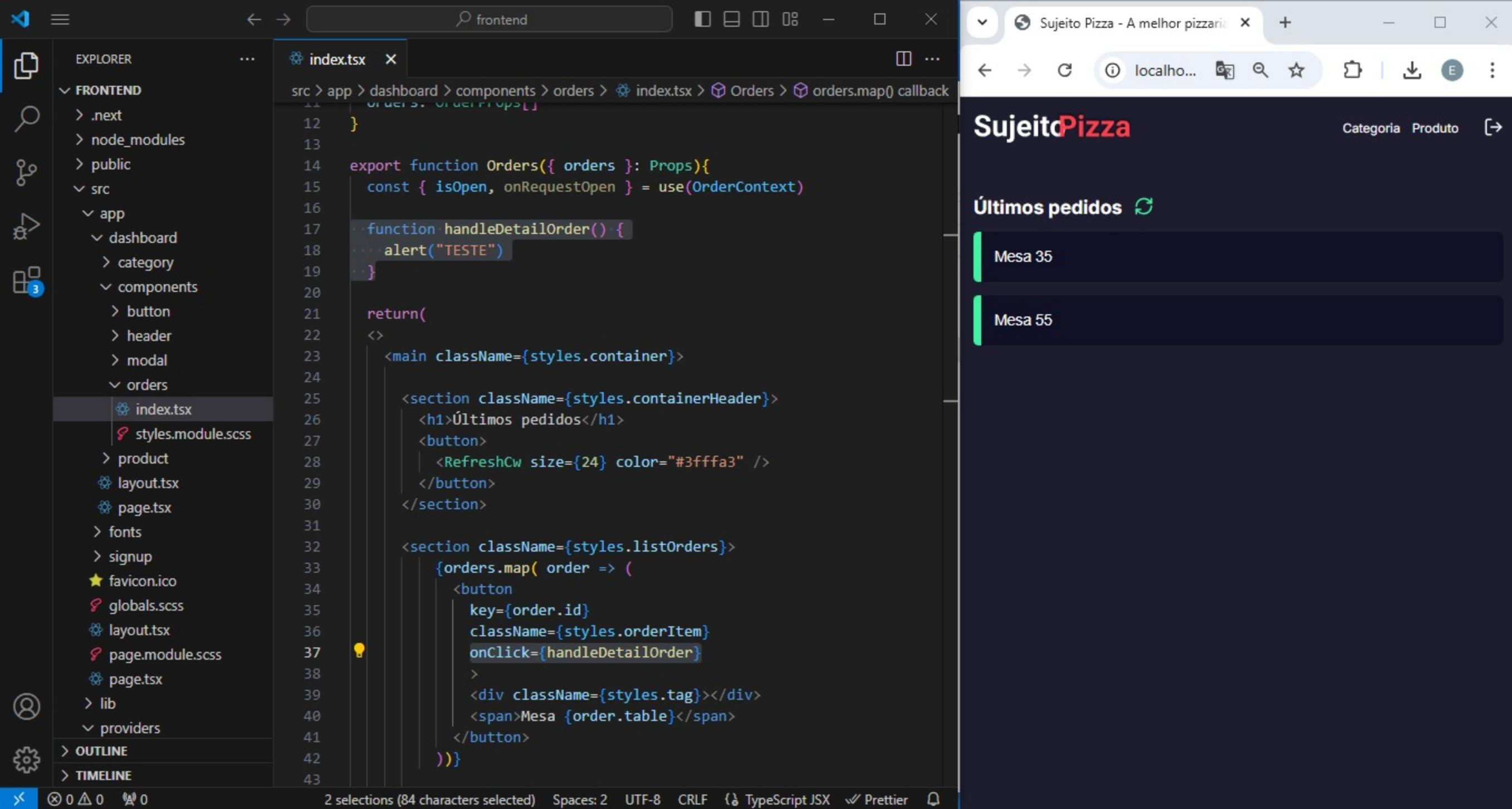
The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file.
- Content:** The code is a TypeScript component named "Orders". It includes imports from "OrderContext", a header section with a refresh button, and a list of orders section. A tooltip for "RefreshCw" is visible over the refresh button.
- Status Bar:** Shows line 19, column 1 (604 selected), spaces: 2, and encoding: UTF-8.

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- Header:** "SujeitoPizza" with navigation links "Categoria" and "Produto".
- Section:** "Últimos pedidos" (Last orders) with a refresh icon.
- List:** Displays two recent orders: "Mesa 35" and "Mesa 55".



The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is open in the editor.
- Content:** The code is a TypeScript component named "Orders". It includes logic for handling order details and rendering a list of orders.

```
src > app > dashboard > components > orders > index.tsx > Orders > orders.map() callback
12 }
13
14 export function Orders({ orders }: Props){
15   const { isOpen, onRequestOpen } = use(OrderContext)
16
17   function handleDetailOrder() {
18     alert("TESTE")
19   }
20
21   return(
22     <>
23       <main className={styles.container}>
24
25         <section className={styles.containerHeader}>
26           <h1>Últimos pedidos</h1>
27           <button>
28             <RefreshCw size={24} color="#3ffffa3" />
29           </button>
30         </section>
31
32         <section className={styles.listOrders}>
33           {orders.map( order => (
34             <button
35               key={order.id}
36               className={styles.orderItem}
37               onClick={handleDetailOrder}
38             >
39               <div className={styles.tag}></div>
40               <span>Mesa {order.table}</span>
41             </button>
42           ))}
43         </section>
44     </main>
45   )
46 }
```

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localhost:3000
- Content:** The page displays a header and a list of tables. An alert box is visible, showing the message "TESTE".

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is open in the editor.
- Content:** The code is a TypeScript component named "Orders". It includes logic for handling order details and rendering a list of orders.

```
src > app > dashboard > components > orders > index.tsx > Orders > orders.map() callback
12 }
13
14 export function Orders({ orders }: Props) {
15   const { isOpen, onRequestOpen } = use(OrderContext)
16
17   function handleDetailOrder() {
18     alert("TESTE")
19   }
20
21   return(
22     <>
23       <main className={styles.container}>
24
25         <section className={styles.containerHeader}>
26           <h1>Últimos pedidos</h1>
27           <button>
28             <RefreshCw size={24} color="#3ffffa3" />
29           </button>
30         </section>
31
32         <section className={styles.listOrders}>
33           {orders.map( order => (
34             <button
35               key={order.id}
36               className={styles.orderItem}
37               onClick={handleDetailOrder}
38             >
39               <div className={styles.tag}></div>
40               <span>Mesa {order.table}</span>
41             </button>
42           ))}
43         </section>
44     </main>
45   )
46 }
```

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localhost:3000
- Content:** The page displays a header with "Últimos pedidos" and a button. Below it, there are two cards, each labeled "Mesa 55". An alert box is overlaid on the screen with the message "TESTE".

frontend

index.tsx

```
src > app > dashboard > components > orders > index.tsx > Orders > handleDetailOrder
```

```
9
10 interface Props{
11   orders: OrderProps[]
12 }
13
14 export function Orders({ orders }: Props){
15   const { isOpen, onRequestOpen } = use(OrderContext)
16
17   function handleDetailOrder() {
18     alert("TESTE")
19   }
20
21   return(
22     <>
23       <main className={styles.container}>
24         <section className={styles.containerHeader}>
25           <h1>Últimos pedidos</h1>
26           <button>
27             <RefreshCw size={24} color="#3fffa3" />
28           </button>
29         </section>
30
31         <section className={styles.listOrders}>
32           {orders.map( order => (
33             <button
34               key={order.id}
35               className={styles.orderItem}
36               onClick={handleDetailOrder}
37             >
38               <div className={styles.tag}></div>
39               <span>Mesa {order.table}</span>
40             
```

Sujeito Pizza - A melhor pizzaria

localhost

SujeitoPizza

Categoria Produto []

Últimos pedidos

- Mesa 35
- Mesa 55

frontend

index.tsx

```
src > app > dashboard > components > orders > index.tsx > Orders > handleDetailOrder

9
10 interface Props{
11   orders: OrderProps[]
12 }
13
14 export function Orders({ orders }: Props){
15   const { isOpen, onRequestOpen } = use(OrderContext)
16
17   function handleDetailOrder() {
18     onRequestOpen()
19   }
20
21   return(
22     <>
23       <main className={styles.container}>
24         <section className={styles.containerHeader}>
25           <h1>Últimos pedidos</h1>
26           <button>
27             <RefreshCw size={24} color="#3fffa3" />
28           </button>
29         </section>
30
31         <section className={styles.listOrders}>
32           {orders.map( order => (
33             <button
34               key={order.id}
35               className={styles.orderItem}
36               onClick={handleDetailOrder}
37             >
38               <div className={styles.tag}></div>
39               <span>Mesa {order.table}</span>
40             
```

Sujeito Pizza - A melhor pizzaria

localhost

SujeitoPizza

Categoria Produto []

Últimos pedidos

- Mesa 35
- Mesa 55

The screenshot shows the VS Code interface with the following details:

- Explorer View:** Shows the project structure under the 'FRONTEND' folder, including 'src', 'app', 'dashboard', 'components', 'orders', 'index.tsx', 'styles.module.scss', 'product', 'layout.tsx', 'page.tsx', 'fonts', 'signup', 'favicon.ico', 'globals.scss', 'layout.tsx', 'page.module.scss', 'page.tsx', 'lib', 'providers', 'order.tsx' (selected), 'services', 'middleware.ts', '.gitignore', 'OUTLINE', and 'TIMELINE'.
- Editor View:** Displays the 'order.tsx' file content in TypeScript. The code defines an 'OrderProvider' component that manages the state of an order request. It includes type definitions for 'OrderProviderProps' and 'OrderContextData', and logic for opening and closing the request state.
- Status Bar:** Shows file statistics: Ln 20, Col 5 (16 selected), Spaces: 2, UTF-8, CRLF, and language settings: TypeScript JSX, Prettier.

The screenshot shows a browser window displaying the 'Sujeito Pizza' website. The page features a dark header with the logo 'SujeitoPizza' and navigation links for 'Categoria' and 'Produto'. Below the header, a section titled 'Últimos pedidos' lists recent orders:

- Mesa 35
- Mesa 55

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under **FRONTEND**. The **orders** folder is selected, containing **index.tsx** and **styles.module.scss**.
- Editor:** The file **index.tsx** is open, displaying TypeScript code for a component named **Orders**. The code includes an interface **Props** with a prop **orders**, a function **handleDetailOrder**, and a return block with sections for header and order list.
- Status Bar:** Shows file statistics: **Ln 22, Col 5**, **Spaces: 2**, **UTF-8**, **CRLF**, **TypeScript JSX**, and **Prettier**.

Browser Preview:

- Title Bar:** **Sujeito Pizza - A melhor pizzaria**
- URL Bar:** **localhost**
- Content:** A dark-themed web page titled "Sujeito Pizza". It displays a header with the logo and some blurred text. Below it is a section titled "Últimos pedidos" with a button labeled "Novo pedido". A modal window is open in the foreground:
 - Header:** **Detalhes do pedido**
 - Text:** Mesa 36
 - List:**
 - 1 - Pizza catupiry**
Pizza de frango com catupiry, borda recheada
 - 3 - Pizza calabresa**
Pizza de frango com catupiry, borda recheada
 - Button:** **Concluir pedido**

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under **FRONTEND**. The **orders** folder is selected, containing **index.tsx** and **styles.module.scss**.
- Editor:** The file **index.tsx** is open, showing TypeScript code for a component named **Orders**. The code includes an interface **Props** with a prop **orders**, a function **handleDetailOrder**, and a return statement with JSX for sections and a map over the orders array.
- Status Bar:** Shows file statistics: **Ln 18, Col 5 (15 selected)**, **Spaces: 2**, **UTF-8**, **CRLF**, **TypeScript JSX**, and **Prettier**.

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localho... (localhost)
- Content:** A dark-themed web page titled "Sujeito Pizza". It displays a header with the logo and some blurred text. Below it, there's a section for "Últimos pedidos" (Last orders) and a detailed view for "Mesa 36".
 - Mesa 36:** Shows two items:
 - 1 - Pizza catupiry:** Pizza de frango com catupiry, borda recheada
 - 3 - Pizza calabresa:** Pizza de frango com catupiry, borda recheada
 - Concluir pedido:** A green button at the bottom right of the detail view.

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "order.tsx" is the active file.
- Content:** The code defines an OrderProvider component using React Context API. It includes type definitions for OrderProviderProps and OrderContextData, and logic for managing the open state of a modal.

```
index.tsx
orders.tsx X

src > providers > order.tsx > OrderProvider
6   onRequestOpen: () => void;
7   onRequestClose: () => void;
8 }
9
10 type OrderProviderProps = {
11   children: ReactNode;
12 }
13
14 export const OrderContext = createContext({} as OrderContextData)
15
16 export function OrderProvider({ children }: OrderProviderProps){
17   const [isOpen, setIsOpen] = useState(false);
18
19   function onRequestOpen(){
20     setIsOpen(true);
21   }
22
23   function onRequestClose(){
24     setIsOpen(false);
25   }
26
27   return(
28     <OrderContext.Provider
29       value={{
30         isOpen,
31         onRequestOpen,
32         onRequestClose
33       }}
34     >
35       {children}
36     </OrderContext.Provider>
37   )
38 }
```

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localho... (localhost)
- Content:** A dark-themed web application showing a modal dialog titled "Detalhes do pedido" (Order Details) for "Mesa 36". The modal lists two pizzas:
 - 1 - Pizza catupiry: Pizza de frango com catupiry, borda recheada
 - 3 - Pizza calabresa: Pizza de frango com catupiry, borda recheadaA green "Concluir pedido" (Finish order) button is at the bottom right.

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under `FRONTEND`. The `orders` folder is selected.
- index.tsx:** The current file is open, showing TypeScript code for rendering a dashboard component. It includes imports from `components` and `orders`, and uses `styled-components` for styling.
- order.tsx:** Another file in the `orders` folder is visible in the background.

```
src > app > dashboard > components > orders > index.tsx > Orders
27
28     <button>
29         <RefreshCw size={24} color="#3ffffa3" />
30     </button>
31 </section>
32 <section className={styles.listOrders}>
33     {orders.map( order => (
34         <button
35             key={order.id}
36             className={styles.orderItem}
37             onClick={handleDetailOrder}
38             >
39                 <div className={styles.tag}></div>
40                 <span>Mesa {order.table}</span>
41             </button>
42     ))}
43 </section>
44 </main>
45
46
47
48     { isOpen && <Modalorder/> }
49
50
51 }
```

Browser Preview:

The browser window displays a dark-themed application for "Sujeito Pizza". A modal dialog is open, titled "Detalhes do pedido". It shows details for an order at Mesa 36:

- 1 - Pizza catupiry:** Pizza de frango com catupiry, borda recheada
- 3 - Pizza calabresa:** Pizza de frango com catupiry, borda recheada

A green button labeled "Concluir pedido" is visible at the bottom right of the modal.

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is open in the editor.
- Content:** The code is for a modal component named "Modalorder". It imports styles from "styles.module.scss", "lucide-react", and "react". It uses the "OrderContext" provider. The component returns a dialog with a close button and a section for order details.

```
import styles from './styles.module.scss'
import { Import, X } from 'lucide-react'
import { use } from 'react'
import { OrderContext } from '@/providers/order'

export function Modalorder(){
    const { onRequestClose } = use(OrderContext);

    return(
        <dialog className={styles.dialogContainer}>
            <section className={styles.dialogContent}>
                <button className={styles.dialogBack}>
                    <X size={40} color="#FF3f4b" />
                </button>

                <article className={styles.container}>
                    <h2>Detalhes do pedido</h2>

                    <span className={styles.table}>
                        Mesa <b>36</b>
                    </span>

                    <section className={styles.item}>
                        <span>1 - <b>Pizza catupiry</b></span>
                        <span className={styles.description}>Pizza de frango com catupiry, borda recheada</span>
                    </section>

                    <section className={styles.item}>
                        <span>3 - <b>Pizza calabresa</b></span>
                        <span className={styles.description}>Pizza de frango com calabresa, borda recheada</span>
                    </section>
                </article>
            </section>
        </dialog>
    )
}
```

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- Header:** Categoria Produto
- Section:** Últimos pedidos
- Content:** Shows the last orders for tables 35 and 55.

| Mesa | Pedidos |
|---------|---------------------|
| Mesa 35 | Pizza catupiry (1) |
| Mesa 55 | Pizza calabresa (3) |

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is open in the editor.
- Content:** The code is for a modal component named "Modalorder". It imports styles from "styles.module.scss", "lucide-react", and "react". It uses the "OrderContext" provider. The component returns a modal dialog with a close button and a section for order details.
- Status Bar:** Shows 2 selections (122 characters selected), Spaces: 2, UTF-8, CRLF, TypeScript JSX, Prettier.

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localho...
- Content:** A modal window titled "Detalhes do pedido" (Order Details) is displayed. It shows an order for "Mesa 36" with two items:
 - 1 - Pizza catupiry**: Pizza de frango com catupiry, borda recheada.
 - 3 - Pizza calabresa**: Pizza de frango com catupiry, borda recheada.A green "Concluir pedido" (Finish order) button is at the bottom.

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under `FRONTEND`. The `modal` folder is expanded, showing `index.tsx` and `styles.module.scss`.
- Editor:** The file `index.tsx` is open in the modal component. The code uses TypeScript and React to render a modal dialog with a close button and a table of pizza details.
- Bottom Status Bar:** Shows file statistics: `Ln 1, Col 1 (12 selected)`, `Spaces: 2`, `UTF-8`, `CRLF`, `TypeScript JSX`, and `Prettier`.

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localho... (localhost)
- Content:** A dark-themed modal window titled "Detalhes do pedido" (Details of the order) for "Mesa 36". It lists two pizzas:
 - 1 - Pizza catupiry**: Pizza de frango com catupiry, borda recheada
 - 3 - Pizza calabresa**: Pizza de frango com catupiry, borda recheadaA green "Concluir pedido" (Finish order) button is at the bottom right.

frontend

index.tsx ... \orders index.tsx ... \modal

src > app > dashboard > components > modal > index.tsx > Modalorder

```
1 "use client"
2
3 import styles from './styles.module.scss'
4 import { Import, X } from 'lucide-react'
5 import { use } from 'react'
6 import { OrderContext } from '@/providers/order'
7
8 export function Modalorder(){
9     const { onRequestClose } = use(OrderContext);
10
11     return(
12         <dialog className={styles.dialogContainer}>
13             <section className={styles.dialogContent}>
14                 <button className={styles.dialogBack} onClick=
15                     {onRequestClose}>
16                     <X size={40} color="#FF3f4b" />
17                 </button>
18
19                 <article className={styles.container}>
20                     <h2>Detalhes do pedido</h2>
21
22                     <span className={styles.table}>
23                         Mesa <b>36</b>
24                     </span>
25
26                     <section className={styles.item}>
27                         <span>1 - <b>Pizza catupiry</b></span>
28                         <span className={styles.description}>Pizza de frango
29                             com catupiry, borda recheada</span>
30                     </section>
31
32                 </article>
33             </section>
34         </dialog>
35     )
36 }
```

Ln 15, Col 71 (24 selected) Spaces: 2 CRLF { TypeScript JSX ✓ Prettier

EXPLORER FRONTEND OUTLINE TIMELINE

Sujeito Pizza - A melhor pizzaria

localhost

SujeitoPizza

Categoria Produto []

Últimos pedidos

Mesa 35

Mesa 55

The image shows a developer's environment with two main windows: a code editor and a browser preview.

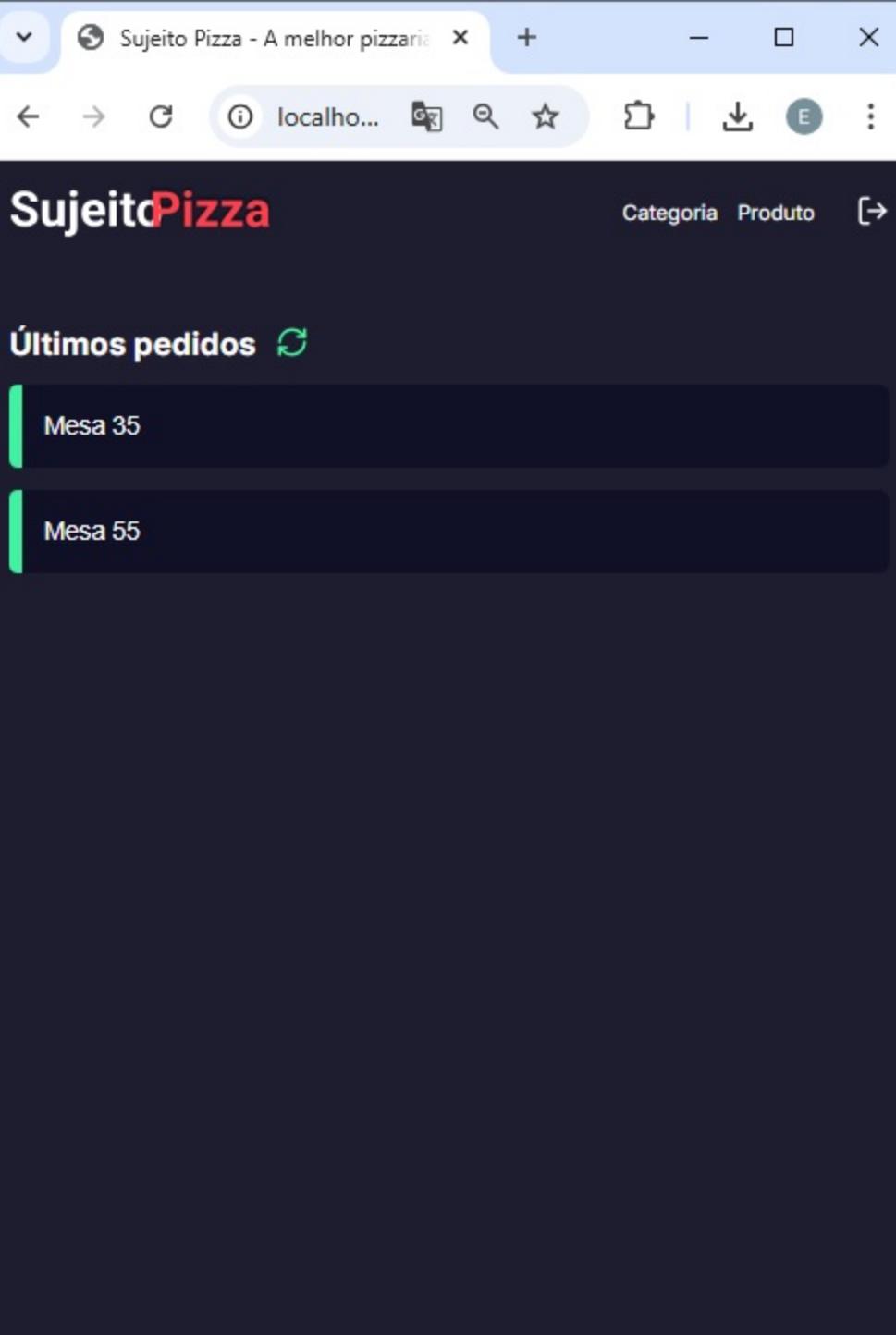
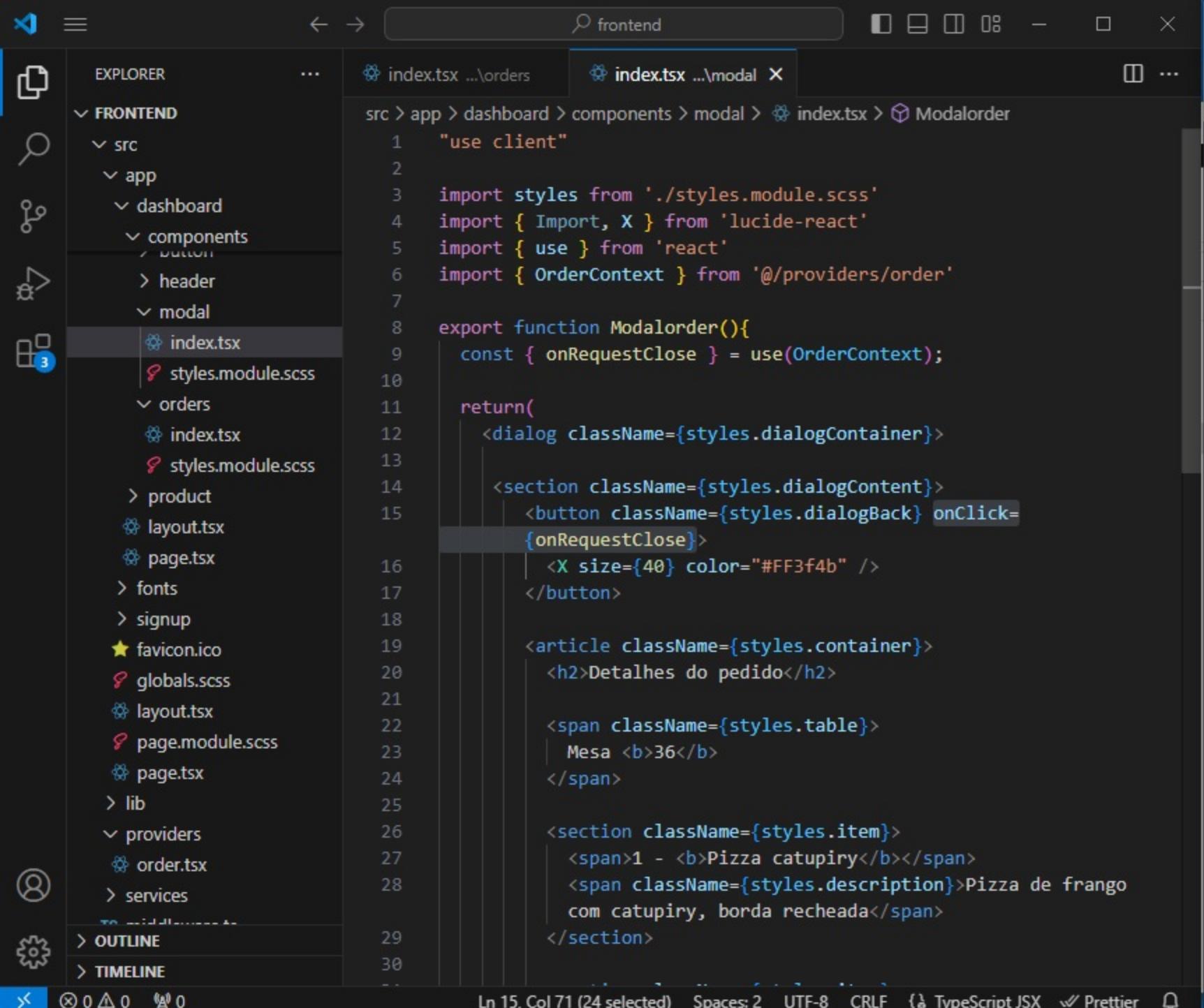
Code Editor (VS Code):

- Explorer:** Shows the project structure under **FRONTEND**. The **modal** folder is selected.
- File:** `index.tsx` (highlighted)
- Content:** TypeScript code for a modal component. The code includes imports for client-side rendering, Lucide icons, React, and a context provider. It defines a function `Modalorder` that returns a modal dialog with a close button and a details section. The details section lists a pizza order for table 36, item 1: "Pizza catupiry" (Chicken with catupiry, filled border).

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- Content:** A modal window titled "Detalhes do pedido" (Order Details) for table 36. It lists the order: "1 - Pizza catupiry" (Chicken with catupiry, filled border). There is also a "Concluir pedido" (Finish order) button.

Bottom status bar: Ln 15, Col 71 (24 selected) Spaces: 2 CRLF { TypeScript JSX ✘ Prettier



The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under **FRONTEND**. The **modal** folder is selected, containing **index.tsx** and **styles.module.scss**.
- Search:** A search bar at the top contains the text "frontend".
- Editor:** The file **index.tsx** is open, showing TypeScript code for a modal component. The code imports styles from **./styles.module.scss**, uses **lucide-react** for icons, and retrieves data from **OrderContext**. It displays a dialog with a close button and details about a pizza order on **Mesa 36**.
- Bottom Status Bar:** Shows the current position as **Ln 15, Col 71 (24 selected)**, encoding as **UTF-8**, and file type as **TypeScript JSX**.

Browser Preview:

- Title Bar:** The browser window title is "Sujeito Pizza - A melhor pizzaria".
- Address Bar:** The address bar shows "localhost" and includes standard browser controls like back, forward, and search.
- Content Area:** Displays the "Sujeito Pizza" logo and a modal dialog titled "Detalhes do pedido" (Details of the order). The dialog shows an order for **Mesa 36** consisting of:
 - 1 - Pizza catupiry**: Pizza de frango com catupiry, borda recheada
 - 3 - Pizza calabresa**: Pizza de frango com catupiry, borda recheadaA green "Concluir pedido" (Finish order) button is visible at the bottom right of the modal.

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is open in the editor.
- Content:** The code is a TypeScript component named "Modalorder". It imports styles from "styles.module.scss", "lucide-react", and "react". It uses the "OrderContext" provider. The component returns a modal dialog with a close button and a section for order details.
- Status Bar:** Shows "Ln 15, Col 71 (24 selected)" and "TypeScript JSX".

Browser Preview:

- Title:** "Sujeito Pizza - A melhor pizzaria"
- Header:** "SujeitoPizza" with navigation links "Categoria" and "Produto".
- Section:** "Últimos pedidos" (Last orders) with a refresh icon.
- List:** Shows two recent orders: "Mesa 35" and "Mesa 55".

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file, located at `src > app > dashboard > components > orders > index.tsx`.
- Content:** The code is a React component named "Orders". It imports various modules and defines a function component "Orders" that renders a header section with a button to handle order details.

```
1  "use client"
2
3  import { use } from 'react'
4  import styles from './styles.module.scss'
5  import { RefreshCw } from 'lucide-react'
6  import { OrderProps } from '@/lib/order.type'
7  import { Modalorder } from '../modal'
8  import { OrderContext } from '@/providers/order'
9
10 interface Props{
11   orders: OrderProps[]
12 }
13
14 export function Orders({ orders }: Props){
15   const { isOpen, onRequestOpen } = use(OrderContext)
16
17   function handleDetailOrder() {
18     onRequestOpen()
19   }
20
21   return(
22     <>
23       <main className={styles.container}>
24         <section className={styles.containerHeader}>
25           <h1>Últimos pedidos</h1>
26           <button>
27             <RefreshCw size={24} color="#3ffffa3" />
28           </button>
29         </section>
30
31         <section className={styles.listOrders}>
32
33           . . .
34
35         </section>
36
37       </main>
38     </>
39   )
40 }
```

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- Header:** "SujeitoPizza" with "Categoria" and "Produto" dropdowns.
- Section:** "Últimos pedidos" with a refresh icon.
- List:** Two items: "Mesa 35" and "Mesa 55".

The image shows a developer's environment with two main windows: a code editor in the foreground and a browser preview in the background.

Code Editor (VS Code):

- Explorer:** Shows the project structure under **FRONTEND**. The **orders** folder in the **app/dashboard/components** directory is selected.
- File:** `index.tsx` (orders)
- Content:** TypeScript code for a component named `Orders`. It imports various modules and defines a function `handleDetailOrder` which calls `onRequestOpen`.

```
1 "use client"
2
3 import { use } from 'react'
4 import styles from './styles.module.scss'
5 import { RefreshCw } from 'lucide-react'
6 import { OrderProps } from '@/lib/order.type'
7 import { Modalorder } from '../modal'
8 import { OrderContext } from '@/providers/order'
9
10 interface Props{
11   orders: OrderProps[]
12 }
13
14 export function Orders({ orders }: Props){
15   const { isOpen, onRequestOpen } = use(OrderContext)
16
17   function handleDetailOrder() {
18     onRequestOpen()
19   }
20
21   return(
22     <>
23       <main className={styles.container}>
24         <section className={styles.containerHeader}>
25           <h1>Últimos pedidos</h1>
26           <button>
27             <RefreshCw size={24} color="#3ffffa3" />
28           </button>
29         </section>
30
31         <section className={styles.listOrders}>
```

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localho... (localhost)
- Content:** A dark-themed web application showing a modal dialog titled "Detalhes do pedido". The modal displays details for "Mesa 36":
 - 1 - Pizza catupiry**: Pizza de frango com catupiry, borda recheada
 - 3 - Pizza calabresa**: Pizza de frango com catupiry, borda recheadaA green button labeled "Concluir pedido" is visible at the bottom right of the modal.

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under **FRONTEND**. The **orders** folder in the **app/dashboard/components** directory is selected.
- Editor:** Displays the **index.tsx** file for the **Orders** component. The code uses TypeScript and React. It imports components like `use client`, `RefreshCw`, and `OrderProps`. The `Orders` component takes an array of `OrderProps` as props and contains logic for handling order details.
- Status Bar:** Shows the current line (Ln 17), column (Col 1), and character count (58 selected). It also indicates the file type as `TypeScript JSX` and the code formatter as `Prettier`.

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localhost:3001
- Content:** The page displays a header with the Sujeito Pizza logo and navigation links. Below it, there's a section titled "Últimos pedidos" (Last orders) featuring a button to refresh the list. A modal window is open, titled "Detalhes do pedido" (Order details) for "Mesa 36". The modal lists two items:
 - 1 - Pizza catupiry**: Pizza de frango com catupiry, borda recheada
 - 3 - Pizza calabresa**: Pizza de frango com catupiry, borda recheadaA green "Concluir pedido" (Finish order) button is at the bottom of the modal.



Search.. Ctrl + P

Invite

Pizzaria



Sujeito Programador ▾

Run

GET http://localhost:3333/order/detail

Send

200 OK

66 ms

1538 B

57 Minutes Ago ▾

Base Environment

Add Cookies

Add Certificates

Filter

order

PUT Finalizar Pedido (Finish Order)

GET Detalhes de um Pedido (Deta... ▾

GET Listar Pedidos (List Orders)

PUT Enviar Pedido (Send Order)

POST Adicionar um Item ao Pedido

DEL Remover Item

POST Criar uma ORDER - Abrir Mesa

DEL Deletar uma ORDER - Pedido

product

POST Cadastrar Produto

GET Listar Produto por Categoria

master



Params 1

Body

Auth

Headers 3

Scripts

Docs

Preview

Headers 8

Cookies

Tests 0 / 0

→ Mock

Console

URL PREVIEW

http://localhost:3333/order/detail?order_id=ee42b24f-cdc5-48fd-87d3-1758a06afae3

QUERY PARAMETERS

Import from URL Bulk Edit

+ Add

Delete all

Description

order_id

ee42b24f-cdc5-48fd-87d3-1758

PATH PARAMETERS

Path parameters are url path segments that start with a colon ':' e.g. ':id'

Preview

```

1 [ 
2   {
3     "id": "1900a07a-65b4-409d-812b-b8aa9a5c7ace",
4     "amount": 2,
5     "created_at": "2024-11-29T02:41:38.002Z",
6     "update_at": "2024-11-29T02:41:38.002Z",
7     "order_id": "ee42b24f-cdc5-48fd-87d3-1758a06afae3",
8     "product_id": "6ad3ded6-115b-468c-89eb-25de1a945944",
9     "product": {
10       "id": "6ad3ded6-115b-468c-89eb-25de1a945944",
11       "name": "Burger Triplo Dev",
12       "price": "50",
13       "description": "Hamburger Triplo com Cheddar",
14       "banner": "9b5565412cabf8912fd21c9c26cfaca6-hamburger
triplo.jpg",
15       "created_at": "2024-11-29T02:14:03.556Z",
16       "update_at": "2024-11-29T02:14:03.556Z",
17       "category_id": "1e4ffc20-04b2-46db-96f0-ceea3ca2975a"
18     },
19     "order": {
20       "id": "ee42b24f-cdc5-48fd-87d3-1758a06afae3",
21       "table": 35,
22       "status": false,
23       "draft": false,
24       "name": "Jorge",
25       "created_at": "2024-11-29T02:28:31.944Z",
26       "update_at": "2024-11-29T02:28:31.944Z"
27     }
28   },

```

\$.store.books[*].author



Preferences

Online

Made with ❤ by Kong

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under **FRONTEND**. The **orders** folder is selected.
- File:** `index.tsx` (orders)
- Content:** TypeScript code for a component named `Orders` that handles order details.

```
1 "use client"
2
3 import { use } from 'react'
4 import styles from './styles.module.scss'
5 import { RefreshCw } from 'lucide-react'
6 import { OrderProps } from '@/lib/order.type'
7 import { Modalorder } from '../modal'
8 import { OrderContext } from '@/providers/order'
9
10 interface Props{
11   orders: OrderProps[]
12 }
13
14 export function Orders({ orders }: Props){
15   const { isOpen, onRequestOpen } = use(OrderContext)
16
17   function handleDetailOrder() {
18     onRequestOpen()
19   }
20
21   return(
22     <>
23       <main className={styles.container}>
24         <section className={styles.containerHeader}>
25           <h1>Últimos pedidos</h1>
26           <button>
27             <RefreshCw size={24} color="#3fffa3" />
28           </button>
29         </section>
30
31         <section className={styles.listOrders}>
```

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localho...
- Content:** A modal window titled "Detalhes do pedido" for "Mesa 36". It lists two items:
 - 1 - Pizza catupiry: Pizza de frango com catupiry, borda recheada
 - 3 - Pizza calabresa: Pizza de frango com catupiry, borda recheadaA green button at the bottom right says "Concluir pedido".

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** OrderProvider.tsx
- Content:** TypeScript code for an OrderProvider component. It includes type definitions, context creation, state management for open status, and provider rendering.

```
index.tsx ..\orders order.tsx index.tsx ..\modal
src > providers > order.tsx > OrderProvider > onRequestOpen
6   onRequestOpen: () => void;
7   onRequestClose: () => void;
8 }
9
10 type OrderProviderProps = {
11   children: ReactNode;
12 }
13
14 export const OrderContext = createContext({} as OrderContextData)
15
16 export function OrderProvider({ children }: OrderProviderProps){
17   const [isOpen, setIsOpen] = useState(false);
18
19   function onRequestOpen(){
20     setIsOpen(true);
21   }
22
23   function onRequestClose(){
24     setIsOpen(false);
25   }
26
27   return(
28     <OrderContext.Provider
29       value={{
30         isOpen,
31         onRequestOpen,
32         onRequestClose
33       }}
34     >
35       {children}
36     </OrderContext.Provider>
37   )
38 }
```

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localho... (localhost)
- Content:** A dark-themed web application interface for a pizza delivery service. It features a header with the logo, a sidebar with navigation links, and a modal window titled "Detalhes do pedido" (Order Details) for "Mesa 36". The modal lists two items: "1 - Pizza catupiry" and "3 - Pizza calabresa", both described as "Pizza de frango com catupiry, borda recheada". A green "Concluir pedido" (Finish order) button is at the bottom.

```
index.tsx ...\\orders order.tsx 1 ✘ index.tsx ...\\modal
src > providers > order.tsx > OrderProvider > onRequestOpen
  6   onRequestOpen: () => void;
  7   onRequestClose: () => void;
  8 }
  9
10 type OrderProviderProps = {
11   children: ReactNode;
12 }
13
14 export const OrderContext = createContext({} as OrderContextData)
15
16 export function OrderProvider({ children }: OrderProviderProps){
17   const [isOpen, setIsOpen] = useState(false);
18
19   function onRequestOpen(order_id: string){
20     console.log(order_id)
21     setIsOpen(true);
22   }
23
24   function onRequestClose(){
25     setIsOpen(false);
26   }
27
28   return(
29     <OrderContext.Provider
30       value={{
31         isOpen,
32         onRequestOpen,
33         onRequestClose
34       }}
35     >
36       {children}
37     </OrderContext.Provider>
38 }
```

SujeitoPizza

Categoria Produto

Últimos pedidos

- Mesa 35
- Mesa 55

The screenshot shows a developer's environment with a code editor and a web browser side-by-side.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "order.tsx" is the active file.
- Content:** The code defines an OrderProvider component using React Context API. It includes type definitions for OrderContextData and OrderProviderProps, and functions for onRequestOpen and onRequestClose.

```
1  "use client"
2  import { createContext, ReactNode, useState } from 'react'
3
4  type OrderContextData = {
5    isOpen: boolean;
6    onRequestOpen: (order_id: string) => void;
7    onRequestClose: () => void;
8  }
9
10 type OrderProviderProps = {
11   children: ReactNode;
12 }
13
14 export const OrderContext = createContext({} as OrderContextData)
15
16 export function OrderProvider({ children }: OrderProviderProps){
17   const [isOpen, setIsOpen] = useState(false);
18
19   function onRequestOpen(order_id: string){
20     console.log(order_id);
21     setIsOpen(true);
22   }
23
24   function onRequestClose(){
25     setIsOpen(false);
26   }
27
28   return(
29     <OrderContext.Provider
30       value={{
31         isOpen,
32         onRequestOpen,
```

Browser:

- Title:** localhost:3000/dashboard
- Error Message:** "Unhandled Runtime Error" - "ChunkLoadError: Loading chunk app/dashboard/layout failed. (timeout: http://localhost:3000/_next/static/chunks/app/dashboard/layout.js)"
- Call Stack:**
 - Next.js
 - <unknown>
 - file:///C/Users/Dev/Desktop/Pizzaria/frontend/.next/static/chunks/webpack.js (153:40)
 - fn.e
 - file:///C/Users/Dev/Desktop/Pizzaria/frontend/.next/static/chunks/webpack.js (389:50)
 - Next.js

The screenshot shows a developer's environment with a code editor and a browser developer tools panel.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** index.tsx (orders) - 1 error.
- Content:** TypeScript code for a component named Orders. The code includes a function handleDetailOrder() which calls onRequestOpen(). The component structure includes sections for the latest orders and individual order items.

```
14  export function Orders({ orders }: Props){  
15    const { isOpen, onRequestOpen } = use(OrderContext)  
16  
17    function handleDetailOrder() {  
18      onRequestOpen()  
19    }  
20  
21    return(  
22      <>  
23        <main className={styles.container}>  
24          <section className={styles.containerHeader}>  
25            <h1>Últimos pedidos</h1>  
26            <button>  
27              <RefreshCw size={24} color="#3ffffa3" />  
28            </button>  
29          </section>  
30  
31          <section className={styles.listOrders}>  
32            {orders.map( order => (  
33              <button  
34                key={order.id}  
35                className={styles.orderItem}  
36                onClick={handleDetailOrder}  
37                >  
38                  <div className={styles.tag}></div>  
39                  <span>Mesa {order.table}</span>  
40                </button>  
41            ))}  
42  
43          </section>  
44  
45        </main>  
46      </>
```

Browser Developer Tools:

- URL:** localhost:3000/dashboard
- Console:** Shows an "Unhandled Runtime Error": "ChunkLoadError: Loading chunk app/dashboard/layout failed. (timeout: http://localhost:3000/_next/static/chunks/app/dashboard/layout.js)".
- Call Stack:**
 - Next.js
 - <unknown>
 - file:///C:/Users/Dev/Desktop/Pizzaria/frontend/.next/static/chunks/webpack.js (153:40)
 - Next.js
 - fn.e
 - file:///C:/Users/Dev/Desktop/Pizzaria/frontend/.next/static/chunks/webpack.js (389:50)
 - Next.js

frontend

EXPLORER

FRONTEND

src

app

dashboard

components

header

modal

index.tsx

styles.module.scss

orders

index.tsx

styles.module.scss

product

layout.tsx

page.tsx

fonts

signup

favicon.ico

globals.scss

layout.tsx

page.module.scss

page.tsx

lib

providers

order.tsx

services

OUTLINE

TIMELINE

index.tsx ..\orders X order.tsx index.tsx ...\\modal

src > app > dashboard > components > orders > index.tsx > Orders > orders.map() callback

```
14  export function Orders({ orders }: Props){  
15    const { isOpen, onRequestOpen } = use(OrderContext)  
16  
17    function handleDetailOrder(order_id: string) {  
18      onRequestOpen(order_id)  
19    }  
20  
21    return(  
22      <>  
23        <main className={styles.container}>  
24          <section className={styles.containerHeader}>  
25            <h1>Últimos pedidos</h1>  
26            <button>  
27              <RefreshCw size={24} color="#3ffffa3" />  
28            </button>  
29          </section>  
30  
31          <section className={styles.listOrders}>  
32            {orders.map( order => (  
33              <button  
34                key={order.id}  
35                className={styles.orderItem}  
36                onClick={() => handleDetailOrder(order.id)}  
37                >  
38                  <div className={styles.tag}></div>  
39                  <span>Mesa {order.table}</span>  
40                </button>  
41            ))}  
42  
43          </section>  
44    </main>  
45  
```

localhost:3000/dashboard

Build Error

Failed to compile

./src/app/dashboard/components/orders/index.tsx

Error: × Expression expected

↳ [c:\Users\Dev\Desktop\Pizzaria\frontend\src\app\dashboard\components\orders\index.tsx:22:1]

19 | }
20 |
21 | return(
22 | <>
23 | <main className={styles.container}>
24 |
24 | <section className={styles.containerHeader}>
25 |
× Unexpected token `button`. Expected jsx identifier

↳ [c:\Users\Dev\Desktop\Pizzaria\frontend\src\app\dashboard\components\orders\index.tsx:34:1]

31 |
32 | <section className={

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file, located at `src > app > dashboard > components > orders > index.tsx`.
- Content:** The code implements a component named "Orders" that displays the last orders. It uses a context provider "OrderContext" and a custom hook "useOrderContext". The component contains a header with a refresh button and a list of orders. Each order item has a "Mais detalhes" button that triggers the "handleDetailOrder" function.

```
interface Props {
  orders: OrderProps[]
}

export function Orders({ orders }: Props) {
  const { isOpen, onRequestOpen } = use(OrderContext)

  function handleDetailOrder(order_id: string) {
    onRequestOpen(order_id)
  }

  return (
    <>
      <main className={styles.container}>
        <section className={styles.containerHeader}>
          <h1>Últimos pedidos</h1>
          <button>
            <RefreshCw size={24} color="#3fffa3" />
          </button>
        </section>

        <section className={styles.listOrders}>
          {orders.map( order => (
            <button
              key={order.id}
              className={styles.orderItem}
              onClick={() => handleDetailOrder(order.id)}
            >
              <div className={styles.tag}></div>
              <span>Mesa {order.table}</span>
            </button>
          ))
        }
      </main>
    </>
  )
}
```

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- Header:** "SujeitoPizza" logo, navigation links for "Categoria" and "Produto", and a search bar.
- Section:** "Últimos pedidos" (Last orders) with a refresh icon.
- Content:** Two items listed: "Mesa 35" and "Mesa 55".

Sujeito Pizza - A melhor pizzaria

localhost:3000/dashboard

Elements Console Sources Network Performance Memory Application Security Default levels No Issues

The resource http://localhost:3000/_next/static/media/logo.77bc50ee.svg was preloaded using link preload but not used within a few seconds from the window's load event. Please make sure it has an appropriate 'as' value and it is preloaded intentionally.

SujeitoPizza

Categoria Produto []

Últimos pedidos

Mesa 35

Mesa 55

Sujeito Pizza - A melhor pizzaria

localhost:3000/dashboard

Elements Console Sources Network Performance Memory Application Security > ▲ 1 ⚙️ ⚙️

Default levels ▾ No Issues ⚙️

The resource http://localhost:3000/_next/static/media/logo.77bc50ee.svg was preloaded using link [preload](#) but not used within a few seconds from the window's load event. Please make sure it has an appropriate 'as' value and it is preloaded intentionally.

ee42b24f-cdc5-48fd-87d3-1758a06afae3 c:\Users\Dev\Desktop\...viders\order.tsx:20

X

Detalhes do pedido

Mesa 36

1 - Pizza catupiry
Pizza de frango com catupiry, borda recheada

3 - Pizza calabresa
Pizza de frango com catupiry, borda recheada

Concluir pedido

Console AI assistance

Sujeito Pizza - A melhor pizzaria X +

localhost:3000/dashboard

SujeitoPizza

Categoria Produto [→]

Últimos pedidos

Mesa 35

Mesa 55

Elements Console Sources Network Performance Memory Application Security > ▲ 1 ⚙️ ⚙️ X

Default levels ▾ No Issues ⚙️

The resource http://localhost:3000/_next/static/media/logo.77bc50ee.svg was preloaded using link [preload](#) but not used within a few seconds from the window's load event. Please make sure it has an appropriate `as` value and it is preloaded intentionally.

ee42b24f-cdc5-48fd-87d3-1758a06afae3 c:\Users\Dev\Desktop...viders\order.tsx:20

Console AI assistance ⚙️

Sujeito Pizza - A melhor pizzaria X +

localhost:3000/dashboard

Elements Console Sources Network Performance Memory Application Security > ▲ 1 | E :

Default levels ▾ No Issues

The resource http://localhost:3000/_next/static/media/logo.77bc50ee.svg was preloaded using link [preload](#) but not used within a few seconds from the window's load event. Please make sure it has an appropriate `as` value and it is preloaded intentionally.

ee42b24f-cdc5-48fd-87d3-1758a06afae3
0d0fef8f-9947-4e0a-94d2-dc2722b9cee0

c:\Users\Dev\Desktop\...viders\order.tsx:20
c:\Users\Dev\Desktop\...viders\order.tsx:20

X

Detalhes do pedido

Mesa 36

1 - **Pizza catupiry**
Pizza de frango com catupiry, borda recheada

3 - **Pizza calabresa**
Pizza de frango com catupiry, borda recheada

Concluir pedido

frontend

EXPLORER

FRONTEND

src

app

dashboard

components

button

header

modal

index.tsx

styles.module.scss

orders

index.tsx

styles.module.scss

product

layout.tsx

page.tsx

fonts

signup

favicon.ico

globals.scss

layout.tsx

page.module.scss

page.tsx

lib

providers

order.tsx

services

modals

OUTLINE

TIMELINE

index.tsx ...\\orders

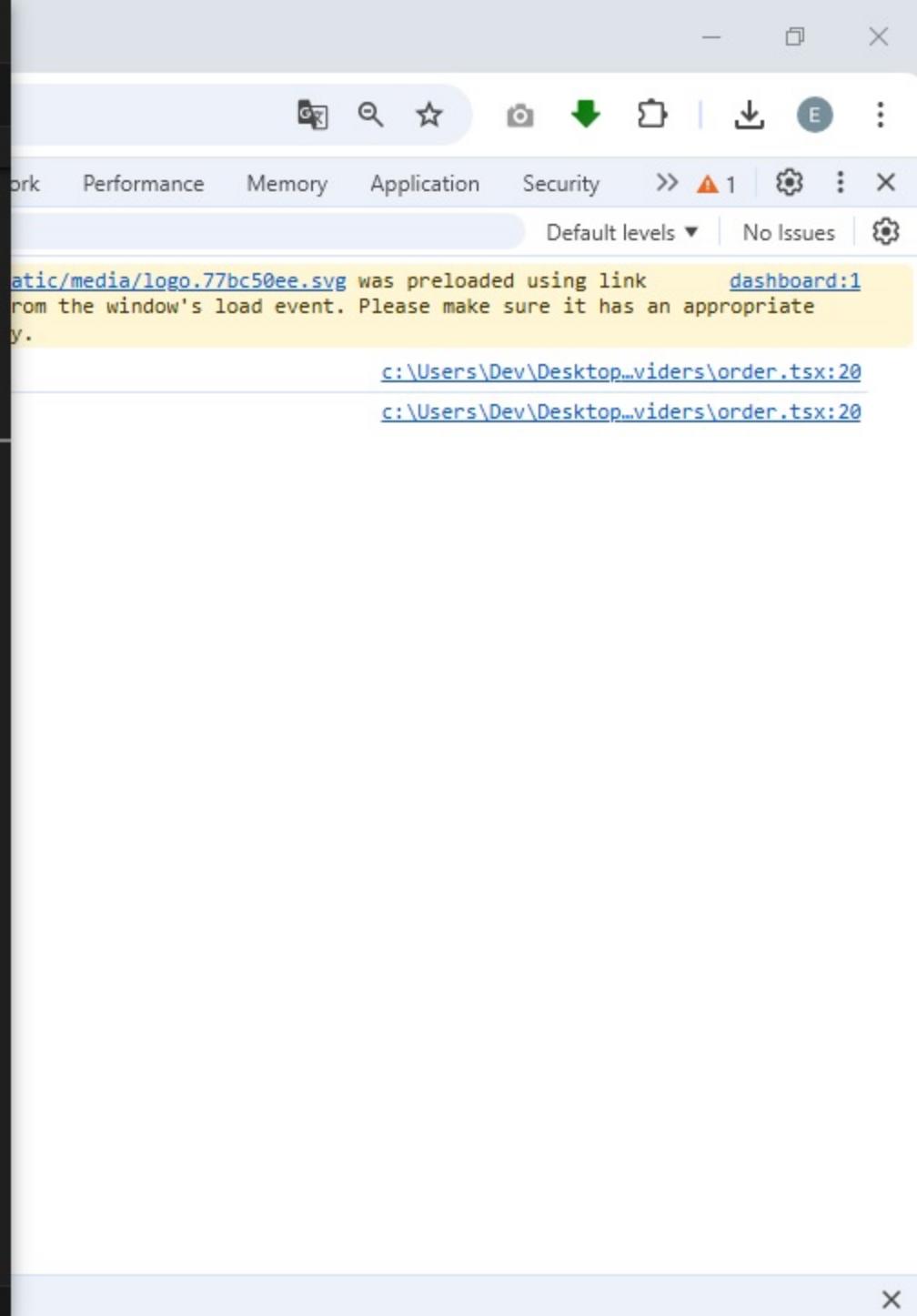
order.tsx

index.tsx ...\\modal

src > providers > order.tsx > OrderProvider > onRequestOpen

```
9
10 type OrderProviderProps = {
11   children: ReactNode;
12 }
13
14 export const OrderContext = createContext({} as OrderContextData)
15
16 export function OrderProvider({ children }: OrderProviderProps){
17   const [isOpen, setIsOpen] = useState(false);
18
19   function onRequestOpen(order_id: string){
20     console.log(order_id);
21     setIsOpen(true);
22   }
23
24   function onRequestClose(){
25     setIsOpen(false);
26   }
27
28   return(
29     <OrderContext.Provider
30       value={{
31         isOpen,
32         onRequestOpen,
33         onRequestClose
34       }}
35     >
36       {children}
37     </OrderContext.Provider>
38   )
39 }
40 }
```

Ln 19, Col 12 (13 selected) Spaces: 2 CRLF TypeScript JSX Prettier



frontend

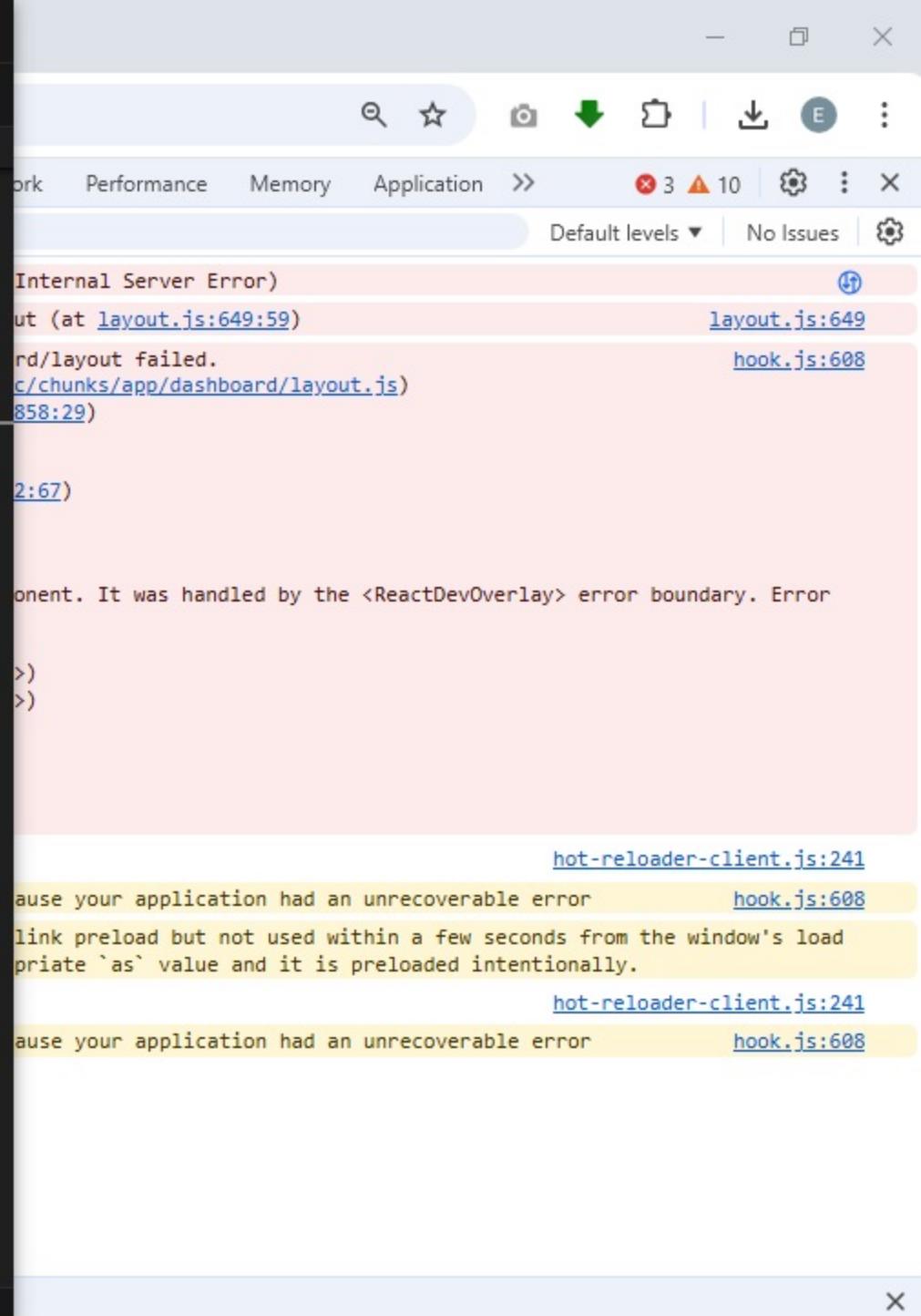
EXPLORER

FRONTEND

src > providers > order.tsx > OrderProvider

```
9
10  type OrderProviderProps = {
11    children: ReactNode;
12  }
13
14  export const OrderContext = createContext({} as OrderContextData)
15
16  export function OrderProvider({ children }: OrderProviderProps){
17    const [isOpen, setIsOpen] = useState(false);
18    const [order, setOrder] = useState([])
19
20    function onRequestOpen(order_id: string){
21      console.log(order_id);
22      setIsOpen(true);
23    }
24
25    function onRequestClose(){
26      setIsOpen(false);
27    }
28
29    return(
30      <OrderContext.Provider
31        value={{
32          isOpen,
33          onRequestOpen,
34          onRequestClose
35        }}
36      >
37        {children}
38      </OrderContext.Provider>
39    )
40 }
```

Ln 18, Col 5 (36 selected) Spaces: 2 UTF-8 CRLF TypeScript JSX Prettier





Search.. Ctrl + P

Invite

Pizzaria



Sujeito Programador ▾

Run

GET http://localhost:3333/order/detail

Send ▾

200 OK

36 ms

1538 B

Just Now ▾

Base Environment

Add Cookies

Add Certificates

Filter

order

PUT Finalizar Pedido (Finish Order)

GET Detalhes de um Pedido (Detail Order)

GET Listar Pedidos (List Orders)

PUT Enviar Pedido (Send Order)

POST Adicionar um Item ao Pedido

DEL Remover Item

POST Criar uma ORDER - Abrir Mesa

DEL Deletar uma ORDER - Pedido

product

POST Cadastrar Produto

GET Listar Produto por Categoria

master



Preferences

Params 1
URL PREVIEW
http://localhost:3333/order/detail?order_id=ee42b24f-cdc5-48fd-87d3-1758a06afae3

Body

Auth

Headers 3

Scripts

Docs

Preview

Headers 8

Cookies

Tests 0 / 0

→ Mock

Console

QUERY PARAMETERS

Import from URL Bulk Edit

+ Add

Delete all

Description

order_id

ee42b24f-cdc5-48fd-87d3-1758

PATH PARAMETERS

Path parameters are url path segments that start with a colon ':' e.g. ':id'

```

1 [ 
2 {
3   "id": "1900a07a-65b4-409d-812b-b8aa9a5c7ace",
4   "amount": 2,
5   "created_at": "2024-11-29T02:41:38.002Z",
6   "update_at": "2024-11-29T02:41:38.002Z",
7   "order_id": "ee42b24f-cdc5-48fd-87d3-1758a06afae3",
8   "product_id": "6ad3ded6-115b-468c-89eb-25de1a945944",
9   "product": {
10     "id": "6ad3ded6-115b-468c-89eb-25de1a945944",
11     "name": "Burger Triplo Dev",
12     "price": "50",
13     "description": "Hamburger Triplo com Cheddar",
14     "banner": "9b5565412cabf8912fd21c9c26cfaca6-hamburger
triplo.jpg",
15     "created_at": "2024-11-29T02:14:03.556Z",
16     "update_at": "2024-11-29T02:14:03.556Z",
17     "category_id": "1e4ffc20-04b2-46db-96f0-ceea3ca2975a"
18   },
19   "order": {
20     "id": "ee42b24f-cdc5-48fd-87d3-1758a06afae3",
21     "table": 35,
22     "status": false,
23     "draft": false,
24     "name": "Jorge",
25     "created_at": "2024-11-29T02:28:31.944Z",
26     "update_at": "2024-11-29T02:28:31.944Z"
27   }
28 },

```

\$store.books[*].author

Online

Made with ❤ by Kong



Search.. Ctrl + P

Invite

Pizzaria



Sujeito Programador ▾

Run

GET http://localhost:3333/order/detail

Send ▾

200 OK

36 ms

1538 B

2 Minutes Ago ▾

Base Environment

Add Cookies

Add Certificates

Filter

order

PUT Finalizar Pedido (Finish Order)

GET Detalhes de um Pedido (Deta... ▾

GET Listar Pedidos (List Orders)

PUT Enviar Pedido (Send Order)

POST Adicionar um Item ao Pedido

DEL Remover Item

POST Criar uma ORDER - Abrir Mesa

DEL Deletar uma ORDER - Pedido

product

POST Cadastrar Produto

GET Listar Produto por Categoria

master



Preview

Headers 8

Cookies

Tests 0 / 0

→ Mock

Console

URL PREVIEW

http://localhost:3333/order/detail?order_id=ee42b24f-cdc5-48fd-87d3-1758a06afae3

QUERY PARAMETERS

Import from URL Bulk Edit

+ Add Delete all Description

order_id

ee42b24f-cdc5-48fd-87d3-1758

PATH PARAMETERS

Path parameters are url path segments that start with a colon ':' e.g. ':id'

Preview ▾

```
1 [
2 {
3   "id": 8
4 },
5 {
6   "id": 8
7 }
8 ]
```

Preferences

Online

Made with ❤ by Kong



Search.. Ctrl + P

Invite

Pizzaria



Sujeito Programador ▾

Run

GET http://localhost:3333/order/detail

Send

200 OK

36 ms

1538 B

3 Minutes Ago ▾

Base Environment

Add Cookies

Add Certificates

Filter

order

PUT Finalizar Pedido (Finish Order)

GET Detalhes de um Pedido (Detail Order)

GET Listar Pedidos (List Orders)

PUT Enviar Pedido (Send Order)

POST Adicionar um Item ao Pedido

DEL Remover Item

POST Criar uma ORDER - Abrir Mesa

DEL Deletar uma ORDER - Pedido

product

POST Cadastrar Produto

GET Listar Produto por Categoria

master



Params 1

Body

Auth

Headers 3

Scripts

Docs

Preview

Headers 8

Cookies

Tests 0 / 0

→ Mock

Console

URL PREVIEW

http://localhost:3333/order/detail?order_id=ee42b24f-cdc5-48fd-87d3-1758a06afae3

QUERY PARAMETERS

Import from URL Bulk Edit

+ Add

Delete all

Description

order_id

ee42b24f-cdc5-48fd-87d3-1758

PATH PARAMETERS

Path parameters are url path segments that start with a colon ':' e.g. ':id'

Preview ▾

```

1  [
2  {
3      "id": "1900a07a-65b4-409d-812b-b8aa9a5c7ace",
4      "amount": 2,
5      "created_at": "2024-11-29T02:41:38.002Z",
6      "update_at": "2024-11-29T02:41:38.002Z",
7      "order_id": "ee42b24f-cdc5-48fd-87d3-1758a06afae3",
8      "product_id": "6ad3ded6-115b-468c-89eb-25de1a945944",
9      "product": {
10         "id": "6ad3ded6-115b-468c-89eb-25de1a945944",
11         "name": "Burger Triplo Dev",
12         "price": "50",
13         "description": "Hamburger Triplo com Cheddar",
14         "banner": "9b5565412cabf8912fd21c9c26cfaca6-hamburger
triplo.jpg",
15         "created_at": "2024-11-29T02:14:03.556Z",
16         "update_at": "2024-11-29T02:14:03.556Z",
17         "category_id": "1e4ffc20-04b2-46db-96f0-ceea3ca2975a"
18     },
19     "order": {
20         "id": "ee42b24f-cdc5-48fd-87d3-1758a06afae3",
21         "table": 35,
22         "status": false,
23         "draft": false,
24         "name": "Jorge",
25         "created_at": "2024-11-29T02:28:31.944Z",
26         "update_at": "2024-11-29T02:28:31.944Z"
27     }
28 },

```

\$.store.books[*].author

Preferences

Online

Made with ❤ by Kong



Search.. Ctrl + P

Invite

Pizzaria ▾



Sujeito Programador ▾

Run

GET ▾ http://localhost:3333/order/detail

Send ▾

200 OK

36 ms

1538 B

6 Minutes Ago ▾

Base Environment

Add Cookies

Add Certificates

Filter

order

PUT Finalizar Pedido (Finish Order)

GET Detalhes de um Pedido (Deta...

GET Listar Pedidos (List Orders)

PUT Enviar Pedido (Send Order)

POST Adicionar um Item ao Pedido

DEL Remover Item

POST Criar uma ORDER - Abrir Mesa

DEL Deletar uma ORDER - Pedido

product

POST Cadastrar Produto

GET Listar Produto por Categoria

master



Params 1

Body

Auth

Headers 3

Scripts

Docs

Preview

Headers 8

Cookies

Tests 0 / 0

→ Mock

Console

URL PREVIEW

http://localhost:3333/order/detail?order_id=ee42b24f-cdc5-48fd-87d3-1758a06afae3

QUERY PARAMETERS

Import from URL Bulk Edit

+ Add

Delete all

Description

order_id

ee42b24f-cdc5-48fd-87d3-1758a06afae3

PATH PARAMETERS

Path parameters are url path segments that start with a colon ':' e.g. ':id'

Preview ▾

```
1 [  
2 {  
3   "id": "1900a07a-65b4-409d-812b-b8aa9a5c7ace",  
4   "amount": 2,  
5   "created_at": "2024-11-29T02:41:38.002Z",  
6   "update_at": "2024-11-29T02:41:38.002Z",  
7   "order_id": "ee42b24f-cdc5-48fd-87d3-1758a06afae3",  
8   "product_id": "6ad3ded6-115b-468c-89eb-25de1a945944",  
9   "product": {  
10     "id": "6ad3ded6-115b-468c-89eb-25de1a945944",  
11     "name": "Burger Triple Dev",  
12     "price": "50",  
13     "description": "Hamburger Triple com Cheddar",  
14     "banner": "9b5565412cabf8912fd21c9c26cfaca6-hamburger  
triple.jpg",  
15     "created_at": "2024-11-29T02:14:03.556Z",  
16     "update_at": "2024-11-29T02:14:03.556Z",  
17     "category_id": "1e4ffc20-04b2-46db-96f0-ceea3ca2975a"  
18   },  
19   "order": {  
20     "id": "ee42b24f-cdc5-48fd-87d3-1758a06afae3",  
21     "table": 35,  
22     "status": false,  
23     "draft": false,  
24     "name": "Jorge",  
25     "created_at": "2024-11-29T02:28:31.944Z",  
26     "update_at": "2024-11-29T02:28:31.944Z"  
27   }  
28 },  
29 ]
```

\$.store.books[*].author

Preferences

Online

Made with ❤ by Kong



Search.. Ctrl + P

Invite

Pizzaria



Sujeito Programador ▾

Run

GET http://localhost:3333/order/detail

Send

200 OK

36 ms

1538 B

7 Minutes Ago ▾

Base Environment

Add Cookies

Add Certificates

Filter

order

PUT Finalizar Pedido (Finish Order)

GET Detalhes de um Pedido (Deta...)

GET Listar Pedidos (List Orders)

PUT Enviar Pedido (Send Order)

POST Adicionar um Item ao Pedido

DEL Remover Item

POST Criar uma ORDER - Abrir Mesa

DEL Deletar uma ORDER - Pedido

product

POST Cadastrar Produto

GET Listar Produto por Categoria

master



Params 1

Body

Auth

Headers 3

Scripts

Docs

Preview

Headers 8

Cookies

Tests 0 / 0

→ Mock

Console

URL PREVIEW

http://localhost:3333/order/detail?order_id=ee42b24f-cdc5-48fd-87d3-1758a06afae3

QUERY PARAMETERS

Import from URL Bulk Edit

+ Add

Delete all

Description

order_id

ee42b24f-cdc5-48fd-87d3-1758

PATH PARAMETERS

Path parameters are url path segments that start with a colon ':' e.g. 'id'

Preview

```

1 [ {
2   "id": "1900a07a-65b4-409d-812b-b8aa9a5c7ace",
3   "amount": 2,
4   "created_at": "2024-11-29T02:41:38.002Z",
5   "update_at": "2024-11-29T02:41:38.002Z",
6   "order_id": "ee42b24f-cdc5-48fd-87d3-1758a06afae3",
7   "product_id": "6ad3ded6-115b-468c-89eb-25de1a945944",
8   "product": {
9     "id": "6ad3ded6-115b-468c-89eb-25de1a945944",
10    "name": "Burger Triple Dev",
11    "price": "50",
12    "description": "A delicious burger with triple meat, cheese, lettuce, and tomato. Served with fries and a drink.",
13    "banner": "9b5e46e6-7fcd-4f3d-843c-520a6-hamburger-triplo.jpg",
14    "created_at": "2024-11-29T02:41:38.002Z",
15    "update_at": "2024-11-29T02:41:38.002Z",
16    "category_id": "6ad3ded6-115b-468c-89eb-25de1a945944",
17  },
18  "order": {
19    "id": "ee42b24f-cdc5-48fd-87d3-1758a06afae3",
20    "table": 35,
21    "status": false,
22    "draft": false,
23    "name": "Jorge",
24    "created_at": "2024-11-29T02:28:31.944Z",
25    "update_at": "2024-11-29T02:28:31.944Z"
26  }
27 },
28 ]

```

Search with Google

Copy

Select All

\$store.books[*].author

Preferences

Online

Made with ❤ by Kong

frontend

EXPLORER

FRONTEND

- src
 - app
 - dashboard
 - components
 - button
 - header
 - modal
 - index.tsx
 - styles.module.scss
 - orders
 - index.tsx
 - styles.module.scss
 - product
 - layout.tsx
 - page.tsx
 - fonts
 - signup
 - lib
 - providers
 - order.tsx
 - services
 - OUTLINE
 - TIMELINE

index.tsx ..\orders

order.tsx

index.tsx ..\modal

```
src > providers > order.tsx > ...
1  "use client"
2  import { createContext, ReactNode, useState } from 'react'
3
4  //-
5  //  "id": "1900a07a-65b4-409d-812b-b8aa9a5c7ace",
6  //  "amount": 2,
7  //  "created_at": "2024-11-29T02:41:38.002Z",
8  //  "update_at": "2024-11-29T02:41:38.002Z",
9  //  "order_id": "ee42b24f-cdc5-48fd-87d3-1758a06afae3",
10 //  "product_id": "6ad3ded6-115b-468c-89eb-25de1a945944",
11 //  "product": {
12 //    "id": "6ad3ded6-115b-468c-89eb-25de1a945944",
13 //    "name": "Burger Triplô Dev",
14 //    "price": 50,
15 //    "description": "Hamburger Triplô com Cheddar",
16 //    "banner": "9b5565412cabf8912fd21c9c26cfaca6-hamburger-triplo.jpg",
17 //    "created_at": "2024-11-29T02:14:03.556Z",
18 //    "update_at": "2024-11-29T02:14:03.556Z",
19 //    "category_id": "1e4ffc20-04b2-46db-96f0-ceea3ca2975a"
20 //  },
21  "order": {
22    "id": "ee42b24f-cdc5-48fd-87d3-1758a06afae3",
23    "table": 35,
24    "status": false,
25    "draft": false,
26    "name": "Jorge",
27    "created_at": "2024-11-29T02:28:31.944Z",
28    "update_at": "2024-11-29T02:28:31.944Z"
29  },
30
31 type OrderContextData = {
```

Ln 4, Col 1 (994 selected) Spaces: 2 UTF-8 CRLF ⓘ TypeScript JSX ✅ Prettier

The screenshot shows a browser window for 'Sujeito Pizza - A melhor pizzaria' at 'localhost:300...'. The main content area displays the 'Últimos pedidos' section with two items: 'Mesa 35' and 'Mesa 55'. The developer tools console tab is active, showing a warning message: 'The resource [dashboard:1](http://localhost:3000/_next/static/media/lo...) was preloaded using link preload but not used within a few seconds from the window's load event. Please make sure it has an appropriate 'as' value and it is preloaded intentionally.' Below the console, there is an 'AI assistance' button.

The image shows a developer's workspace with three main panels: a code editor, a browser preview, and a terminal.

Code Editor: The left panel displays the file `order.tsx` from a project structure named "FRONTEND". The code defines an interface `OrderItemProps` with properties for an item and an order.

```
use client
import { createContext, ReactNode, useState } from 'react'

interface OrderItemProps{
  id: string;
  amount: number;
  created_at: string;
  order_id: string;
  product_id: string;
  product:{
    id: string;
    name: string;
    price: string;
    description: string;
    banner: string;
    category_id: string;
  };
  order:{
    id: string;
    table: number;
    name: string;
    draft: boolean;
    status: boolean;
  }
}

// {
//   "id": "1900a07a-65b4-409d-812b-b8aa9a5c7ace",
//   "amount": 2,
//   "created_at": "2024-11-29T02:41:38.002Z",
//   "update_at": "2024-11-29T02:41:38.002Z",
//   "order_id": "ee42b24f-cdc5-48fd-87d3-1758a06afae3",
// }
```

Browser Preview: The middle panel shows a dark-themed web application titled "Sujeito Pizza - A melhor pizzaria". It features a header with the title and a section titled "Últimos pedidos" displaying two cards: "Mesa 35" and "Mesa 55".

Terminal: The right panel is a terminal window titled "Console" showing logs. It includes a warning message about a preloaded resource that was not used within a few seconds of the window's load event.

```
The resource http://localhost:3000/_next/static/media/lo... was preloaded using link preload but not used within a few seconds from the window's load event. Please make sure it has an appropriate 'as' value and it is preloaded intentionally.
```

The screenshot shows a code editor interface with the following details:

- Top Bar:** Includes icons for back, forward, search, and file operations, followed by the title "frontend".
- Left Sidebar (EXPLORER):** Shows the project structure under "FRONTEND".
 - src:** Contains "app", "dashboard", "components", "fonts", "layout.tsx", "page.tsx", "product", "signup", "favicon.ico", "globals.scss", "layout.tsx", "page.module.scss", "page.tsx", "lib", "providers", "order.tsx" (highlighted), "services", and "utils".
- Current File:** "order.tsx" is the active file, showing code related to "OrderContextData".
- Code Preview:** The code includes a JSON-like object definition and a type declaration for "OrderContextData".

```
25     }
26
27     /**
28      * id: "1900a07a-65b4-409d-812b-b8aa9a5c7ace",
29      * amount: 2,
30      * created_at: "2024-11-29T02:41:38.002Z",
31      * update_at: "2024-11-29T02:41:38.002Z",
32      * order_id: "ee42b24f-cdc5-48fd-87d3-1758a06afae3",
33      * product_id: "6ad3ded6-115b-468c-89eb-25de1a945944",
34      * product: {
35        id: "6ad3ded6-115b-468c-89eb-25de1a945944",
36        name: "Burger Triple Dev",
37        price: "50",
38        description: "Hamburger Triple com Cheddar",
39        banner: "9b5565412cabf8912fd21c9c26cfaca6-hamburger-triplo.jpg",
40        created_at: "2024-11-29T02:14:03.556Z",
41        update_at: "2024-11-29T02:14:03.556Z",
42        category_id: "1e4ffc20-04b2-46db-96f0-ceea3ca2975a"
43      },
44      order: {
45        id: "ee42b24f-cdc5-48fd-87d3-1758a06afae3",
46        table: 35,
47        status: false,
48        draft: false,
49        name: "Jorge",
50        created_at: "2024-11-29T02:28:31.944Z",
51        update_at: "2024-11-29T02:28:31.944Z"
52      },
53
54      type OrderContextData = {
55        isOpen: boolean;
```

- Bottom Status Bar:** Shows file navigation (0, 0, 0, 0), code statistics (Spaces: 2, CRLF), and toolbars (TypeScript JSX, Prettier).

The image shows a developer's workspace with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under **FRONTEND**, including **src**, **app**, **dashboard**, **orders**, **product**, **layout.tsx**, **page.tsx**, **fonts**, **signup**, **favicon.ico**, **globals.scss**, **layout.tsx**, **page.module.scss**, **page.tsx**, **lib**, **providers** (with **order.tsx** selected), **services**, **outline**, and **timeline**.
- Editor:** Displays the file **order.tsx** containing TypeScript code for defining an **OrderItemProps** interface and an **OrderContextData** type.

```
use client
import { createContext, ReactNode, useState } from 'react'

interface OrderItemProps{
  id: string;
  amount: number;
  created_at: string;
  order_id: string;
  product_id: string;
  product:{
    id: string;
    name: string;
    price: string;
    description: string;
    banner: string;
    category_id: string;
  };
  order:{
    id: string;
    table: number;
    name: string;
    draft: boolean;
    status: boolean;
  }
}

type OrderContextData = {
  isOpen: boolean;
  onRequestOpen: (order_id: string) => void;
  onRequestClose: () => void;
}
```

Browser Preview:

The browser shows the application running at **localhost:3001**. The page title is **Sujeito Pizza - A melhor pizzaria**. The main content area displays the heading **SujeitoPizza** and a section titled **Últimos pedidos** with two items: **Mesa 35** and **Mesa 55**.

The image shows a developer's workspace with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under **FRONTEND**, including **src**, **app**, **dashboard**, **components**, **header**, **modal**, **index.tsx**, **styles.module.scss**, **orders**, **index.tsx**, **styles.module.scss**, **product**, **layout.tsx**, **page.tsx**, **fonts**, **signup**, **favicon.ico**, **globals.scss**, **layout.tsx**, **page.module.scss**, **page.tsx**, **lib**, **providers** (with **order.tsx** selected), **services**, **OUTLINE**, and **TIMELINE**.
- Order.tsx Content:**

```
src > providers > order.tsx > OrderProvider
  type OrderContextData = {
    isOpen: boolean;
    onRequestOpen: (order_id: string) => void;
    onRequestClose: () => void;
  }

  type OrderProviderProps = {
    children: ReactNode;
  }

  export const OrderContext = createContext({} as OrderContextData)

  export function OrderProvider({ children }: OrderProviderProps){
    const [isOpen, setIsOpen] = useState(false);
    const [order, setOrder] = useState<OrderItemProps[]>([])

    function onRequestOpen(order_id: string){
      console.log(order_id);
      setIsOpen(true);
    }

    function onRequestClose(){
      setIsOpen(false);
    }

    return(
      <OrderContext.Provider
        value={{
          isOpen,
          onRequestOpen,
          onRequestClose
        }}
    )
  }
}

Ln 41, Col 37 (18 selected) Spaces: 2 CRLF TypeScript JSX Prettier
```

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localhost:3000
- Content:** The page displays the "Últimos pedidos" section with two items: "Mesa 35" and "Mesa 55".
- Console:** Shows a warning message about a preloaded resource not being used within a few seconds of the window's load event.

Bottom status bar: **Ln 41, Col 37 (18 selected)** **Spaces: 2** **CRLF** **TypeScript JSX** **Prettier**

The image shows a developer's workspace with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under **FRONTEND**, including **src**, **app**, **dashboard**, **components**, **orders**, **product**, **layout.tsx**, **page.tsx**, **fonts**, **signup**, **favicon.ico**, **globals.scss**, **layout.tsx**, **page.module.scss**, **page.tsx**, **lib**, **providers**, and **order.tsx**.
- order.tsx:** The current file being edited contains TypeScript code defining an interface **OrderItemProps** and a type **OrderContextData**. The code uses `useClient` from react, imports `createContext` and `useState` from react, and imports `api` from '@services/api'.

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localhost:3000
- Content:** The page displays the logo "SujeitoPizza" and a section titled "Últimos pedidos" (Last orders) showing two items: "Mesa 35" and "Mesa 55".
- Console:** Shows a warning message: "The resource http://localhost:3000/_next/static/media/lo... was preloaded using link preload but not used within a few seconds from the window's load event. Please make sure it has an appropriate 'as' value and it is preloaded intentionally."

At the bottom of the screen, there are status bars for the code editor and browser, showing file paths, line counts, and other system information.

The screenshot shows a developer's environment with two main windows: a code editor in the foreground and a browser preview in the background.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "order.tsx" is the active file, displaying TypeScript code for a context provider.
- Code:**

```
src > providers > order.tsx > OrderProvider > onRequestOpen
35 |     children: ReactNode;
36 |
37
38 export const OrderContext = createContext({} as OrderContextData)
39
40 export function OrderProvider({ children }: OrderProviderProps){
41     const [isOpen, setIsOpen] = useState(false);
42     const [order, setOrder] = useState<OrderItemProps>([])

43     async function onRequestOpen(order_id: string){
44         // console.log(order_id);
45         setIsOpen(true);
46     }

47     function onRequestClose(){
48         setIsOpen(false);
49     }

50     return(
51         <OrderContext.Provider
52             value={{
53                 isOpen,
54                 onRequestOpen,
55                 onRequestClose
56             }}
57         >
58             {children}
59         </OrderContext.Provider>
60     )
61 }
62
63
64 }
```
- Status Bar:** Shows "2 selections (30 characters selected)" and various file encoding and format settings.

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localhost:3000
- Content:** Displays the "Últimos pedidos" section of the application, listing "Mesa 35" and "Mesa 55".
- Console:** Shows browser logs and Fast Refresh-related messages.
- Messages:** Lists several warnings about Fast Refresh and resource preloading.

The screenshot shows a VS Code interface with the following details:

- Explorer View:** On the left, the tree view shows the project structure. The `src` folder contains `app`, `orders`, and `providers`. The `providers` folder is expanded, showing `order.tsx` (marked with a red dot and a '1' icon), `services`, and `types`.
- Search Bar:** At the top center, it says "frontend".
- Code Editor:** The main area displays the content of `order.tsx`. The code defines an `OrderContext` context provider and an `OrderProvider` component. It includes an `onRequestOpen` function that logs the `order_id` and makes an API call to get order details using `getCookieClient` to get a token.
- Status Bar:** At the bottom, it shows "Ln 47, Col 1 (262 selected)" and "Spaces: 2".
- Bottom Icons:** Icons for "TypeScript JSX" and "Prettier" are visible at the bottom right.

```
index.tsx ...\\orders
order.tsx 1 X
index.tsx ...\\modal

src > providers > order.tsx > OrderProvider > onRequestOpen
35     children: ReactNode;
36 }
37
38 export const OrderContext = createContext({} as OrderContextData)
39
40 export function OrderProvider({ children }: OrderProviderProps){
41     const [isOpen, setIsOpen] = useState(false);
42     const [order, setOrder] = useState<OrderItemProps[]>([])
43
44     async function onRequestOpen(order_id: string){
45         // console.log(order_id);
46
47         const token = getCookieClient();
48
49         const response = await api.get("/order/detail", {
50             headers: {
51                 Authorization: `Bearer ${token}`
52             },
53             params: {
54                 order_id: order_id
55             }
56         })
57
58         console.log(response.data);
59
60         setIsOpen(true);
61     }
62
63     function onRequestClose(){
64         setIsOpen(false);
65     }
66
67     return children;
}
```

```
use client"
import { createContext, ReactNode, useState } from 'react'
import { api } from '@/services/api';
import { getCookieClient } from '@/lib/cookieClient';

interface OrderItemProps{
  id: string;
  amount: number;
  created_at: string;
  order_id: string;
  product_id: string;
  product:{
    id: string;
    name: string;
    price: string;
    description: string;
    banner: string;
    category_id: string;
  };
  order:{
    id: string;
    table: number;
    name: string;
    draft: boolean;
    status: boolean;
  }
}

type OrderContextData = {
  isOpen: boolean;
  onRequestOpen: (order_id: string) => void;
  onRequestClose: () => void;
}
```

SujeitoPizza

Últimos pedidos

- Mesa 35
- Mesa 55

Sujeito Pizza - A melhor pizzaria X +

localhost:3000/dashboard

SujeitoPizza

Categoria Produto [→]

Últimos pedidos

Mesa 35

Mesa 55

Elements Console Sources Network Performance Memory Application Security > ▲ 1 | E :

The resource http://localhost:3000/_next/static/media/logo.77bc50ee.svg was preloaded using link preload but not used within a few seconds from the window's load event. Please make sure it has an appropriate 'as' value and it is preloaded intentionally.

>

Console Ai assistance

Sujeito Pizza - A melhor pizzaria

localhost:3000/dashboard

Elements Console Sources Network Performance Memory Application Security > ▲ 2 | Default levels | No Issues

The resource http://localhost:3000/_next/static/media/logo.77bc50ee.svg was preloaded using link [dashboard:1](#) preload but not used within a few seconds from the window's load event. Please make sure it has an appropriate 'as' value and it is preloaded intentionally.

▶ (2) [...], [...]

c:\Users\Dev\Desktop\...viders\order.tsx:59

The resource http://localhost:3000/_next/static/media/logo.77bc50ee.svg was preloaded using link [dashboard:1](#) preload but not used within a few seconds from the window's load event. Please make sure it has an appropriate 'as' value and it is preloaded intentionally.

>

Detalhes do pedido

Mesa 36

1 - Pizza catupiry
Pizza de frango com catupiry, borda recheada

3 - Pizza calabresa
Pizza de frango com catupiry, borda recheada

Concluir pedido

Sujeito Pizza - A melhor pizzaria X +

localhost:3000/dashboard

Elements Console Sources Network Performance Memory Application Security > ▲ 2 | Settings | More

Default levels ▾ | No Issues

The resource http://localhost:3000/_next/static/media/logo.77bc50ee.svg was preloaded using link [dashboard:1](#) preload but not used within a few seconds from the window's load event. Please make sure it has an appropriate 'as' value and it is preloaded intentionally.

c:\Users\Dev\Desktop\viders\order.tsx:59

▼ (2) [{}], [{}]

▶ 0: {id: '1900a07a-65b4-409d-812b-b8aa9a5c7ace', amount: 2, created_at: '2024-11-29T02:41:38.002Z', update_at: '2024-12-03T23:07:45.492Z', length: 2}

▶ 1: {id: '471a49ae-7db2-4dd8-8f6d-476040a6e166', amount: 1, created_at: '2024-12-03T23:07:45.492Z', update_at: ''}

▶ [[Prototype]]: Array(0)

The resource http://localhost:3000/_next/static/media/logo.77bc50ee.svg was preloaded using link [dashboard:1](#) preload but not used within a few seconds from the window's load event. Please make sure it has an appropriate 'as' value and it is preloaded intentionally.

>

X

Detalhes do pedido

Mesa 36

1 - Pizza catupiry
Pizza de frango com catupiry, borda recheada

3 - Pizza calabresa
Pizza de frango com catupiry, borda recheada

Concluir pedido

Sujeito Pizza - A melhor pizzaria X +

localhost:3000/dashboard

Elements Console Sources Network Performance Memory Application Security > ▲ 2 ⚙️ ⚙️ X

Default levels ▾ No Issues

The resource http://localhost:3000/_next/static/media/logo.77bc50ee.svg was preloaded using link [dashboard:1](#) preload but not used within a few seconds from the window's load event. Please make sure it has an appropriate 'as' value and it is preloaded intentionally.

c:\Users\Dev\Desktop...viders\order.tsx:59

▼ (2) [{}], {}] ⓘ
▼ 0:
amount: 2
created_at: "2024-11-29T02:41:38.002Z"
id: "1900a07a-65b4-409d-812b-b8aa9a5c7ace"
► order: {id: 'ee42b24f-cdc5-48fd-87d3-1758a06afae3', table: 35, status: false, draft: false, name: 'Jorge', ...}
order_id: "ee42b24f-cdc5-48fd-87d3-1758a06afae3"
► product: {id: '6ad3ded6-115b-468c-89eb-25de1a945944', name: 'Burger Triplo Dev', price: '50', description: 'H...', product_id: "6ad3ded6-115b-468c-89eb-25de1a945944"
update_at: "2024-11-29T02:41:38.002Z"
► [[Prototype]]: Object
► 1: {id: '471a49ae-7db2-4dd8-8f6d-476040a6e166', amount: 1, created_at: '2024-12-03T23:07:45.492Z', update_at: '...', length: 2
► [[Prototype]]: Array(0)

The resource http://localhost:3000/_next/static/media/logo.77bc50ee.svg was preloaded using link [dashboard:1](#) preload but not used within a few seconds from the window's load event. Please make sure it has an appropriate 'as' value and it is preloaded intentionally.

>

Console AI assistance

Detalhes do pedido

Mesa 36

1 - **Pizza catupiry**
Pizza de frango com catupiry, borda recheada

3 - **Pizza calabresa**
Pizza de frango com catupiry, borda recheada

Concluir pedido

Sujeito Pizza - A melhor pizzaria

localhost:3000/dashboard

Elements Console Sources Network Performance Memory Application Security > ▲ 2 | No Issues |

The resource http://localhost:3000/_next/static/media/logo.77bc50ee.svg was preloaded using link [dashboard:1](#) preload but not used within a few seconds from the window's load event. Please make sure it has an appropriate 'as' value and it is preloaded intentionally.

c:\Users\Dev\Desktop\...viders\order.tsx:59

```
▼ (2) [...], [...] ⓘ
  ▼ 0:
    amount: 2
    created_at: "2024-11-29T02:41:38.002Z"
    id: "1900a07a-65b4-409d-812b-b8aa9a5c7ace"
    ▼ order:
      created_at: "2024-11-29T02:28:31.944Z"
      draft: false
      id: "ee42b24f-cdc5-48fd-87d3-1758a06afae3"
      name: "Jorge"
      status: false
      table: 35
      update_at: "2024-11-29T02:28:31.944Z"
      ► [[Prototype]]: Object
      order_id: "ee42b24f-cdc5-48fd-87d3-1758a06afae3"
      ► product: {id: '6ad3ded6-115b-468c-89eb-25de1a945944', name: 'Burger Triple Dev', price: '50', description: 'H'}
        product_id: "6ad3ded6-115b-468c-89eb-25de1a945944"
        update_at: "2024-11-29T02:41:38.002Z"
        ► [[Prototype]]: Object
      ► 1: {id: '471a49ae-7db2-4dd8-8f6d-476040a6e166', amount: 1, created_at: '2024-12-03T23:07:45.492Z', update_at: '}
        length: 2
      ► [[Prototype]]: Array(0)
```

The resource http://localhost:3000/_next/static/media/logo.77bc50ee.svg was preloaded using link [dashboard:1](#) preload but not used within a few seconds from the window's load event. Please make sure it has an appropriate 'as' value and it is preloaded intentionally.

Concluir pedido

Sujeito Pizza - A melhor pizzaria X +

localhost:3000/dashboard

Elements Console Sources Network Performance Memory Application Security > ▲ 2 | Settings | More

The resource http://localhost:3000/_next/static/media/logo.77bc50ee.svg was preloaded using link `preload` but not used within a few seconds from the window's load event. Please make sure it has an appropriate `'as'` value and it is preloaded intentionally.

c:\Users\Dev\Desktop\viders\order.tsx:59

```
▼ (2) [{}], {}]
  ▼ 0:
    amount: 2
    created_at: "2024-11-29T02:41:38.002Z"
    id: "1900a07a-65b4-409d-812b-b8aa9a5c7ace"
    ▼ order:
      created_at: "2024-11-29T02:28:31.944Z"
      draft: false
      id: "ee42b24f-cdc5-48fd-87d3-1758a06afae3"
      name: "Jorge"
      status: false
      table: 35
      update_at: "2024-11-29T02:28:31.944Z"
      ► [[Prototype]]: Object
      order_id: "ee42b24f-cdc5-48fd-87d3-1758a06afae3"
    ▼ product:
      banner: "9b5565412cabf8912fd21c9c26cfaca6-hamburger triplo.jpg"
      category_id: "1e4ffc20-04b2-46db-96f0-ceea3ca2975a"
      created_at: "2024-11-29T02:14:03.556Z"
      description: "Hamburger Triplo com "
      id: "6ad3ded6-115b-468c-89eb-25de1a945944"
      name: "Burger Triplo Dev"
      price: "50"
      update_at: "2024-11-29T02:14:03.556Z"
      ► [[Prototype]]: Object
      product_id: "6ad3ded6-115b-468c-89eb-25de1a945944"
      update_at: "2024-11-29T02:41:38.002Z"
      ► [[Prototype]]: Object
    ▶ 1: {id: '471a49ae-7db2-4dd8-8f6d-476040a6e166', amount: 1, created_at: '2024-12-03T23:07:45.492Z', update_at: '2024-11-29T02:41:38.002Z', length: 2}
    ► [[Prototype]]: Array(0)
```

X

Detalhes do pedido

Mesa 36

1 - Pizza catupiry
Pizza de frango com catupiry, borda recheada

3 - Pizza calabresa
Pizza de frango com catupiry, borda recheada

Concluir pedido

Sujeito Pizza - A melhor pizzaria X +

localhost:3000/dashboard

Elements Console Sources Network Performance Memory Application Security > ▲ 2 | Settings | More

The resource http://localhost:3000/_next/static/media/logo.77bc50ee.svg was preloaded using link `preload` but not used within a few seconds from the window's load event. Please make sure it has an appropriate `'as'` value and it is preloaded intentionally.

c:\Users\Dev\Desktop\...viders\order.tsx:59

```
▼ (2) [{}], {}]
  ► 0: {id: '1900a07a-65b4-409d-812b-b8aa9a5c7ace', amount: 2, created_at: '2024-11-29T02:41:38.002Z', update_at: '2024-12-03T23:07:45.492Z', order: {created_at: '2024-11-29T02:28:31.944Z', draft: false, id: 'ee42b24f-cdc5-48fd-87d3-1758a06afae3', name: 'Jorge', status: false, table: 35, update_at: '2024-11-29T02:28:31.944Z'}, product: {banner: 'c83f6cc7a8b3dea100c14366e0808e99-hamburger.jpg', category_id: '1e4ffc20-04b2-46db-96f0-ceea3ca2975a', created_at: '2024-11-29T01:06:24.050Z', description: 'Burgão Dev de Salada com 1 hambúrguer de 160g', id: '96eb4477-824b-4137-8251-dbce1d72f755', name: 'Dev Burger Salada', price: '40', update_at: '2024-11-29T01:06:24.050Z'}}, 1: {amount: 1, created_at: '2024-12-03T23:07:45.492Z', id: '471a49ae-7db2-4dd8-8f6d-476040a6e166', order: {created_at: '2024-11-29T02:28:31.944Z', draft: false, id: 'ee42b24f-cdc5-48fd-87d3-1758a06afae3', name: 'Jorge', status: false, table: 35, update_at: '2024-11-29T02:28:31.944Z'}, product: {banner: 'c83f6cc7a8b3dea100c14366e0808e99-hamburger.jpg', category_id: '1e4ffc20-04b2-46db-96f0-ceea3ca2975a', created_at: '2024-11-29T01:06:24.050Z', description: 'Burgão Dev de Salada com 1 hambúrguer de 160g', id: '96eb4477-824b-4137-8251-dbce1d72f755', name: 'Dev Burger Salada', price: '40', update_at: '2024-11-29T01:06:24.050Z'}}]
```

X

Detalhes do pedido

Mesa 36

1 - Pizza catupiry
Pizza de frango com catupiry, borda recheada

3 - Pizza calabresa
Pizza de frango com catupiry, borda recheada

Concluir pedido

Sujeito Pizza - A melhor pizzaria

localhost:3000/dashboard

SujeitoPizza

Categoria Produto [→]

Últimos pedidos

Mesa 35

Mesa 55

Elements Console Sources Network Performance Memory Application Security Default levels No Issues

Filter

The resource http://localhost:3000/_next/static/media/logo.77bc50ee.svg was preloaded using link [dashboard:1](#) preload but not used within a few seconds from the window's load event. Please make sure it has an appropriate 'as' value and it is preloaded intentionally.

▶ (2) [...], [...]

c:\Users\Dev\Desktop...viders\order.tsx:59

The resource http://localhost:3000/_next/static/media/logo.77bc50ee.svg was preloaded using link [dashboard:1](#) preload but not used within a few seconds from the window's load event. Please make sure it has an appropriate 'as' value and it is preloaded intentionally.

>

Console AI assistance

Sujeito Pizza - A melhor pizzaria X +

localhost:3000/dashboard

SujeitoPizza

Categoria Produto [→]

Últimos pedidos

Mesa 35

Mesa 55

Elements Console Sources Network Performance Memory Application Security > ▲ 2 | ⚙ | ... X

Default levels ▾ | No Issues | ⚙

The resource http://localhost:3000/_next/static/media/logo.77bc50ee.svg was preloaded using link [dashboard:1](#) preload but not used within a few seconds from the window's load event. Please make sure it has an appropriate `as` value and it is preloaded intentionally.

▶ (2) [...], [...]

c:\Users\Dev\Desktop...viders\order.tsx:59

The resource http://localhost:3000/_next/static/media/logo.77bc50ee.svg was preloaded using link [dashboard:1](#) preload but not used within a few seconds from the window's load event. Please make sure it has an appropriate `as` value and it is preloaded intentionally.

>

Clear console

Clear console history

Save as...

Console AI assistance

Sujeito Pizza - A melhor pizzaria

localhost:3000/dashboard

SujeitoPizza

Categoria Produto [→]

Últimos pedidos

Mesa 35

Mesa 55

Elements Console Sources Network Performance Memory Application Security > Default levels ▾ No Issues

Console AI assistance

Sujeito Pizza - A melhor pizzaria X +

localhost:3000/dashboard

Elements Console Sources Network Performance Memory Application Security > ▲ 1 | E :

Default levels ▾ No Issues

▶ (2) [..., ...]

⚠ The resource http://localhost:3000/_next/static/media/logo.77bc50ee.svg was preloaded using link [preload](#) but not used within a few seconds from the window's load event. Please make sure it has an appropriate `as` value and it is preloaded intentionally.

c:\Users\Dev\Desktop...viders\order.tsx:59

X

Detalhes do pedido

Mesa 36

1 - Pizza catupiry
Pizza de frango com catupiry, borda recheada

3 - Pizza calabresa
Pizza de frango com catupiry, borda recheada

Concluir pedido

Sujeito Pizza - A melhor pizzaria

localhost:3000/dashboard

Detalhes do pedido

Mesa 36

1 - Pizza catupiry
Pizza de frango com catupiry, borda recheada

3 - Pizza calabresa
Pizza de frango com catupiry, borda recheada

Concluir pedido

Elements Console Sources Network Performance Memory Application Security Default levels No Issues

c:\Users\Dev\Desktop...viders\order.tsx:59

(2) [{}], {}

0:

amount: 2
created_at: "2024-11-26T14:14:15.760Z"
id: "1fc66991-1e7e-42a7-a75e-217ccb149c4d"
order: {id: '0d0fef8f-9947-4e0a-94d2-dc2722b9cee0', table: 55, status: false, draft: false, name: null, ...}
order_id: "0d0fef8f-9947-4e0a-94d2-dc2722b9cee0"
product:
banner: "765f54ffa7f05a9ff902a1a4b85f0e3b-coca-cola lata 350ml.jpg"
category_id: "00d8f79f-5536-429a-bd59-7a82f18700ac"
created_at: "2024-11-25T22:17:10.368Z"
description: "Coca-Cola lata 350ml"
id: "ca5c64fa-e4dd-4b67-9c33-2c4961d4b0b1"
name: "Coca lata"
price: "5"
update_at: "2024-11-25T22:17:10.368Z"
[[Prototype]]: Object
product_id: "ca5c64fa-e4dd-4b67-9c33-2c4961d4b0b1"
update_at: "2024-11-26T14:14:15.760Z"
[[Prototype]]: Object
1: {id: '0d94f740-43ba-4b43-9663-001465523545', amount: 3, created_at: '2024-11-26T17:09:39.369Z', update_at: '
length: 2
[[Prototype]]: Array(0)

The resource http://localhost:3000/_next/static/media/logo.77bc50ee.svg was preloaded using link [dashboard:1](#) preload but not used within a few seconds from the window's load event. Please make sure it has an appropriate 'as' value and it is preloaded intentionally.

Console AI assistance

Sujeito Pizza - A melhor pizzaria

localhost:3000/dashboard

Detalhes do pedido

Mesa 36

1 - Pizza catupiry
Pizza de frango com catupiry, borda recheada

3 - Pizza calabresa
Pizza de frango com catupiry, borda recheada

Concluir pedido

Elements Console Sources Network Performance Memory Application Security Default levels No Issues

```
created_at: "2024-11-26T14:14:15.760Z"
id: "1fc66991-1e7e-42a7-a75e-217ccb149c4d"
▶ order: {id: '0d0fef8f-9947-4e0a-94d2-dc2722b9cee0', table: 55, status: false, draft: false, name: null, ...}
  order_id: "0d0fef8f-9947-4e0a-94d2-dc2722b9cee0"
  ▶ product:
    banner: "765f54ffa7f05a9ff902a1a4b85f0e3b-coca-cola lata 350ml.jpg"
    category_id: "00d8f79f-5536-429a-bd59-7a82f18700ac"
    created_at: "2024-11-25T22:17:10.368Z"
    description: "Coca-Cola lata 350ml"
    id: "ca5c64fa-e4dd-4b67-9c33-2c4961d4b0b1"
    name: "Coca lata"
    price: "5"
    update_at: "2024-11-25T22:17:10.368Z"
    ▶ [[Prototype]]: Object
    product_id: "ca5c64fa-e4dd-4b67-9c33-2c4961d4b0b1"
    update_at: "2024-11-26T14:14:15.760Z"
    ▶ [[Prototype]]: Object
    ▶ 1:
      amount: 3
      created_at: "2024-11-26T17:09:39.369Z"
      id: "0d94f740-43ba-4b43-9663-001465523545"
      ▶ order: {id: '0d0fef8f-9947-4e0a-94d2-dc2722b9cee0', table: 55, status: false, draft: false, name: null, ...}
        order_id: "0d0fef8f-9947-4e0a-94d2-dc2722b9cee0"
        ▶ product:
          banner: "8a652b025acaa467024b0a74085c90f9-pizza de calabresa.jpg"
          category_id: "e21422b1-f695-4426-aa06-9d23c2f3502b"
          created_at: "2024-11-25T22:06:31.016Z"
          description: "Pizza de calabresa saborosa"
          id: "7f66d79a-ceb2-4175-b4a6-dc5208a9f50e"
          name: "Pizza de calabresa"
          price: "46"
          update_at: "2024-11-25T22:06:31.016Z"
          ▶ [[Prototype]]: Object
          product_id: "7f66d79a-ceb2-4175-b4a6-dc5208a9f50e"
          update_at: "2024-11-26T17:09:39.369Z"
```

Sujeito Pizza - A melhor pizzaria

localhost:3000/dashboard

Console

Elements Sources Network Performance Memory Application Security Default levels No Issues

(2) [{}]

The resource http://localhost:3000/_next/static/media/logo.77bc50ee.svg was preloaded using link [preload](#) but not used within a few seconds from the window's load event. Please make sure it has an appropriate 'as' value and it is preloaded intentionally.

Sujeito Pizza

Detalhes do pedido

Mesa 36

1 - Pizza catupiry
Pizza de frango com catupiry, borda recheada

3 - Pizza calabresa
Pizza de frango com catupiry, borda recheada

Concluir pedido

UC: Projetos

Assunto:

Projeto Pizzaria – Frontend_02 (NextJS)

Sistema Pizzaria Web

61 – Exibindo Modal na Tela

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- order.tsx:** The active file contains TypeScript code for a context provider. It defines an `OrderProviderProps` type and an `OrderContext`. The `OrderProvider` component uses `useState` to manage the state of `isOpen` and `order`. It includes an asynchronous function `onRequestOpen` that makes a API call to get order details and logs the response. It also includes a `onRequestClose` function to set `isOpen` back to false.

```
src > providers > order.tsx > OrderProvider > onRequestOpen
35  type OrderProviderProps = {
36    children: ReactNode;
37  }
38
39  export const OrderContext = createContext({} as OrderContextData)
40
41  export function OrderProvider({ children }: OrderProviderProps){
42    const [isOpen, setIsOpen] = useState(false);
43    const [order, setOrder] = useState<OrderItemProps[]>([])
44
45    async function onRequestOpen(order_id: string){
46      // console.log(order_id);
47
48      const token = getCookieClient();
49
50      const response = await api.get("/order/detail", {
51        headers: {
52          Authorization: `Bearer ${token}`
53        },
54        params: {
55          order_id: order_id
56        }
57      });
58
59      console.log(response.data);
60
61      setIsOpen(true);
62    }
63
64    function onRequestClose(){
65      setIsOpen(false);
66    }
67}
```

Browser Preview:

The browser window displays a dark-themed pizza delivery application. The title bar says "Sujeito Pizza - A melhor pizzaria". The main content area shows a table of orders. A modal dialog is open for "Mesa 36", listing two pizzas:

- 1 - Pizza catupiry**: Pizza de frango com catupiry, borda recheada
- 3 - Pizza calabresa**: Pizza de frango com catupiry, borda recheada

A green "Concluir pedido" button is at the bottom right of the modal.

The image shows a developer's environment with two main windows: a code editor in the foreground and a browser preview in the background.

Code Editor (VS Code):

- Explorer:** Shows the project structure with a file named `order.tsx` selected under the `providers` folder in the `src` directory.
- Search:** A search bar at the top contains the text `frontend`.
- Editor:** The `order.tsx` file is open, displaying TypeScript code for a `OrderProvider`. The code handles order details, token retrieval, and state management for an order context.
- Bottom Status Bar:** Shows file information (Ln 59, Col 28), code style settings (Spaces: 2, CRLF), and Prettier status.

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localho... (highlighted in blue)
- Content:** Displays the "Últimos pedidos" (Last orders) section of the website, listing "Mesa 35" and "Mesa 55".
- Header:** Includes links for `Categoria`, `Produto`, and a search icon.

The image shows a developer's environment with two main windows: a code editor in the foreground and a browser preview in the background.

Code Editor (VS Code):

- Explorer:** Shows the project structure with a file named `order.tsx` selected in the providers folder.
- order.tsx Content:** Displays TypeScript code for a context provider. It defines `OrderContextData` and `OrderProviderProps` types, and implements the `OrderProvider` component which manages an open order and provides order details via an API call.

```
order.tsx
src > providers > order.tsx > OrderContextData
19
20   order: {
21     id: string;
22     table: number;
23     name: string;
24     draft: boolean;
25     status: boolean;
26   }
27 }
28
29 type OrderContextData = {
30   isOpen: boolean;
31   onRequestOpen: (order_id: string) => void;
32   onRequestClose: () => void;
33 }
34
35 type OrderProviderProps = {
36   children: ReactNode;
37 }
38
39 export const OrderContext = createContext({} as OrderContextData)
40
41 export function OrderProvider({ children }: OrderProviderProps){
42   const [isOpen, setIsOpen] = useState(false);
43   const [order, setOrder] = useState<OrderItemProps>([])
44
45   async function onRequestOpen(order_id: string){
46     // console.log(order_id);
47
48     const token = getCookieClient();
49
50     const response = await api.get("/order/detail", {
51       headers: {
```

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- Content:** Shows the "Últimos pedidos" (Last orders) section with two items: "Mesa 35" and "Mesa 55".

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure with files like `.next`, `node_modules`, `public`, `src/app`, `src/lib`, and `src/providers/order.tsx`.
- order.tsx Content:**

```
src > providers > order.tsx > OrderContextData
19
20   order: {
21     id: string;
22     table: number;
23     name: string;
24     draft: boolean;
25     status: boolean;
26   }
27 }
28
29 type OrderContextData = {
30   isOpen: boolean;
31   onRequestOpen: (order_id: string) => Promise<void>;
32   onRequestClose: () => void;
33 }
34
35 type OrderProviderProps = {
36   children: ReactNode;
37 }
38
39 export const OrderContext = createContext({} as OrderContextData)
40
41 export function OrderProvider({ children }: OrderProviderProps){
42   const [isOpen, setIsOpen] = useState(false);
43   const [order, setOrder] = useState<OrderItemProps[]>([])
44
45   async function onRequestOpen(order_id: string){
46     // console.log(order_id);
47
48     const token = getCookieClient();
49
50     const response = await api.get("/order/detail", {
51       headers: {
```

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- Header:** Categoria | Produto | ↗
- Section:** Últimos pedidos
- Items:** Mesa 35, Mesa 55

The browser preview shows a dark-themed website for "Sujeito Pizza" displaying the last orders for tables 35 and 55.

The screenshot shows the VS Code interface with the following details:

- Explorer View:** Shows the project structure under "FRONTEND".
- Search Bar:** Contains the text "frontend".
- Editor Tabs:** Two tabs are open: "order.tsx" and "index.tsx".
- Code Editor:** Displays the content of "index.tsx".
- Bottom Status Bar:** Shows "2 selections (10 characters selected)", "Spaces: 2", "UTF-8", "CRLF", "TypeScript JSX", "Prettier", and a refresh icon.

```
src > app > dashboard > components > orders > index.tsx > Orders > handleDetailOrder
  1  "use client"
  2
  3  import { use } from 'react'
  4  import styles from './styles.module.scss'
  5  import { RefreshCw } from 'lucide-react'
  6  import { OrderProps } from '@/lib/order.type'
  7  import { Modalorder } from '../modal'
  8  import { OrderContext } from '@/providers/order'
  9
 10 interface Props{
 11   orders: OrderProps[]
 12 }
 13
 14 export function Orders({ orders }: Props){
 15   const { isOpen, onRequestOpen } = use(OrderContext)
 16
 17   async function handleDetailOrder(order_id: string) {
 18     await onRequestOpen(order_id)
 19   }
 20
 21   return(
 22     <>
 23       <main className={styles.container}>
 24         <section className={styles.containerHeader}>
 25           <h1>Últimos pedidos</h1>
 26           <button>
 27             <RefreshCw size={24} color="#3ffffa3" />
 28           </button>
 29         </section>
 30
 31         <section className={styles.listOrders}>
 32
 33           . . .
 34
 35         </section>
 36
 37       </main>
 38     </>
 39   )
 40 }
```

The screenshot shows a web browser window displaying the "Sujeito Pizza" website. The page content includes:

- Header:** "SujeitoPizza" logo, "Categoria", "Produto", and a search bar.
- Section:** "Últimos pedidos" with two items: "Mesa 35" and "Mesa 55".

frontend

EXPLORER

FRONTEND

- src
- app
- dashboard
 - product
 - layout.tsx
 - page.tsx
 - fonts
 - signup
- favicon.ico
- globals.scss
- layout.tsx
- page.module.scss
- page.tsx
- lib
- providers
 - order.tsx 1
 - services
 - middleware.ts
- .gitignore
- next-env.d.ts
- next.config.ts
- package-lock.json
- package.json
- README.md
- tsconfig.json

OUTLINE

TIMELINE

order.tsx 1 X index.tsx

src > providers > order.tsx > OrderContextData

```
25     status: boolean;
26 }
27 }
28
29 type OrderContextData = {
30   isOpen: boolean;
31   onRequestOpen: (order_id: string) => Promise<void>;
32   onRequestClose: () => void;
33   order: OrderItemProps[];
34 }
35
36 type OrderProviderProps = {
37   children: ReactNode;
38 }
39
40 export const OrderContext = createContext({} as OrderContextData)
41
42 export function OrderProvider({ children }: OrderProviderProps){
43   const [isOpen, setIsOpen] = useState(false);
44   const [order, setOrder] = useState<OrderItemProps[]>([])
45
46   async function onRequestOpen(order_id: string){
47     // console.log(order_id);
48
49     const token = getCookieClient();
50
51     const response = await api.get("/order/detail", {
52       headers: {
53         Authorization: `Bearer ${token}`
54       },
55       params: {
56         order_id: order_id
57       }
58     })
59
60     setOrder(response.data);
61   }
62
63   return (
64     <OrderContext.Provider value={{ isOpen, onRequestOpen, order }}>
65       {children}
66     </OrderContext.Provider>
67   )
68 }
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
```

Ln 33, Col 3 (24 selected) Spaces: 2 CRLF TypeScript JSX Prettier

Sujeito Pizza - A melhor pizzaria

localhost

SujeitoPizza

Categoria Produto []

Últimos pedidos

- Mesa 35
- Mesa 55

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under **FRONTEND**, including **src**, **app**, **dashboard**, **product**, **layout.tsx**, **page.tsx**, **fonts**, **signup**, **favicon.ico**, **globals.scss**, **layout.module.scss**, **page.module.scss**, **page.tsx**, **lib**, **providers** (selected), **order.tsx** (selected), **services**, **middleware.ts**, **.gitignore**, **next-env.d.ts**, **next.config.ts**, **package-lock.json**, **package.json**, **README.md**, and **tsconfig.json**.
- order.tsx:** The currently selected file contains the following TypeScript code:

```
function onRequestClose() {
  setIsOpen(false);
}

return(
  <OrderContext.Provider
    value={{
      isOpen,
      onRequestOpen,
      onRequestClose,
      order
    }}
  >
  {children}
  </OrderContext.Provider>
)
```

Browser Preview:

The browser window displays the application "Sujeito Pizza - A melhor pizzaria". The page shows the header "SujeitoPizza" and a section titled "Últimos pedidos" (Last orders) which lists "Mesa 35" and "Mesa 55".

Bottom Status Bar:

Ln 74, Col 14 (16 selected) Spaces: 2 UTF-8 CRLF TypeScript JSX Prettier

frontend

EXPLORER

FRONTEND

- > .next
- > node_modules
- > public
- src
 - app
 - dashboard
 - > category
 - > components
 - > button
 - > header
 - > modal
 - > index.tsx
 - > styles.module.scss
 - orders
 - > index.tsx
 - > styles.module.scss
 - > product
 - > layout.tsx
 - > page.tsx
 - > fonts
 - > signup
 - favicon.ico
 - > globals.scss
 - > layout.tsx
 - > page.module.scss
 - > page.tsx
- > OUTLINE
- > TIMELINE

order.tsx

index.tsx ... \modal X

index.tsx ... \orders

Ln 9, Col 32 (7 selected) Spaces: 2 UTF-8 CRLF ⓘ TypeScript JSX ✅ Prettier

```
1 "use client"
2
3 import styles from './styles.module.scss'
4 import { Import, X } from 'lucide-react'
5 import { use } from 'react'
6 import { OrderContext } from '@/providers/order'
7
8 export function Modalorder(){
9   const { onRequestClose, order } = use(OrderContext);
10
11   return(
12     <dialog className={styles.dialogContainer}>
13       <section className={styles.dialogContent}>
14         <button className={styles.dialogBack} onClick={onRequestClose}>
15           <X size={40} color="#FF3f4b" />
16         </button>
17
18         <article className={styles.container}>
19           <h2>Detalhes do pedido</h2>
20
21           <span className={styles.table}>
22             Mesa <b>36</b>
23           </span>
24
25           <section className={styles.item}>
26             <span>1 - <b>Pizza catupiry</b></span>
27             <span className={styles.description}>Pizza de frango
28               com catupiry, borda recheada</span>
29           </section>
30         </article>
31       </section>
32     </dialog>
33   )
34 }
```

Sujeito Pizza - A melhor pizzaria

localhost

SujeitoPizza

Categoria Produto []

Últimos pedidos

Mesa 35

Mesa 55

frontend

EXPLORER

FRONTEND

- > .next
- > node_modules
- > public
- src
 - app
 - dashboard
 - > category
 - > components
 - > button
 - > header
 - > modal
 - index.tsx
 - styles.module.scss
 - orders
 - index.tsx
 - styles.module.scss
 - product
 - layout.tsx
 - page.tsx
 - fonts
 - signup
 - favicon.ico
 - globals.scss
 - layout.tsx
 - page.module.scss
 - page.tsx

OUTLINE

TIMELINE

order.tsx

index.tsx ... \modal X

index.tsx ... \orders

Ln 23, Col 21 (2 selected) Spaces: 2 UTF-8 CRLF ⓘ TypeScript JSX ✅ Prettier

```
import { Import, X } from 'lucide-react'
import { use } from 'react'
import { OrderContext } from '@/providers/order'

export function Modalorder(){
    const { onRequestClose, order } = use(OrderContext);

    return(
        <dialog className={styles.dialogContainer}>
            <section className={styles.dialogContent}>
                <button className={styles.dialogBack} onClick={onRequestClose}>
                    <X size={40} color="#FF3f4b" />
                </button>

                <article className={styles.container}>
                    <h2>Detalhes do pedido</h2>

                    <span className={styles.table}>
                        Mesa <b>36</b>
                    </span>

                    <section className={styles.item}>
                        <span>1 - <b>Pizza catupiry</b></span>
                        <span className={styles.description}>Pizza de frango com catupiry, borda recheada</span>
                    </section>

                    <section className={styles.item}>
                        <span>3 - <b>Pizza calabresa</b></span>
                        <span className={styles.description}>Pizza de frango</span>
                    </section>
                </article>
            </section>
        </dialog>
    )
}
```

Sujeito Pizza - A melhor pizzaria

localhost

Sujeito Pizza

Categoria Produto

Últimos pedidos

Mesa 35

Mesa 55

frontend

EXPLORER

FRONTEND

- > .next
- > node_modules
- > public
- src
 - app
 - dashboard
 - > category
 - > components
 - > button
 - > header
 - > modal
 - index.tsx
 - styles.module.scss
 - orders
 - index.tsx
 - styles.module.scss
 - product
 - layout.tsx
 - page.tsx
 - fonts
 - signup
 - favicon.ico
 - globals.scss
 - layout.tsx
 - page.module.scss
 - page.tsx

OUTLINE

TIMELINE

order.tsx index.tsx ...\\modal X index.tsx ...\\orders

src > app > dashboard > components > modal > index.tsx > Modalorder

```
4 import { Import, X } from 'lucide-react'
5 import { use } from 'react'
6 import { OrderContext } from '@/providers/order'

8 export function Modalorder(){
9   const { onRequestClose, order } = use(OrderContext);

11   return(
12     <dialog className={styles.dialogContainer}>
13       <section className={styles.dialogContent}>
14         <button className={styles.dialogBack} onClick={onRequestClose}>
15           <X size={40} color="#FF3f4b" />
16         </button>

19         <article className={styles.container}>
20           <h2>Detalhes do pedido</h2>

23             <span className={styles.table}>
24               Mesa <b>{order[0].order.table}</b>
25             </span>

28               <section className={styles.item}>
29                 <span>1 - <b>Pizza catupiry</b></span>
30                 <span className={styles.description}>Pizza de frango
31                   com catupiry, borda recheada</span>
32               </section>

35               <section className={styles.item}>
36                 <span>3 - <b>Pizza calabresa</b></span>
37                 <span className={styles.description}>Pizza de frango
38                   com calabresa, borda recheada</span>
39               </section>
40             </article>
41           </dialog>
42         </div>
43       </div>
44     </div>
45   </div>
46 </div>
```

Ln 23, Col 21 (22 selected) Spaces: 2 UTF-8 CRLF ↵ TypeScript JSX ✅ Prettier

Sujeito Pizza - A melhor pizzaria

localhost...

SujeitoPizza

Categoria Produto

Últimos pedidos

Mesa 35

Mesa 55

The image shows a developer's environment with two main windows: a code editor in the foreground and a browser preview in the background.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file.
- Content:** The code is a modal component for displaying order details. It imports "lucide-react" for icons, "react", and "OrderContext". It uses the Context API to access the order. The component displays the order details and allows closing the modal.
- Bottom Status Bar:** Shows "Ln 23, Col 21 (22 selected)" and other file statistics.

Browser Preview:

- Title:** "Sujeito Pizza - A melhor pizzaria"
- Content:** A modal window titled "Detalhes do pedido" (Order Details) is displayed. It shows an order for "Mesa 35" with two items:
 - 1 - Pizza catupiry**: Pizza de frango com catupiry, borda recheada
 - 3 - Pizza calabresa**: Pizza de frango com catupiry, borda recheada
- Buttons:** A green "Concluir pedido" (Finish order) button is at the bottom right of the modal.

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- Search:** Shows search results for "frontend".
- Frontend:** The current file is "index.tsx" located at `src > app > dashboard > components > modal > index.tsx`. The code defines a modal component named "Modalorder". It imports "lucide-react" for icons, "react" for useState, and "OrderContext" from a provider. The component returns a dialog with a close button and an article section containing details about an order. The order data is fetched from `order[0].order.table`.
- Outline:** Shows the outline of the current file.
- Timeline:** Shows the history of changes made to the file.

Browser Preview:

The browser window displays a dark-themed application for "Sujeito Pizza - A melhor pizzaria". A modal dialog is open with the title "Detalhes do pedido" (Details of the order). It shows an order for "Mesa 55" (Table 55) consisting of two pizzas:

- 1 - Pizza catupiry**: Pizza de frango com catupiry, borda recheada.
- 3 - Pizza calabresa**: Pizza de frango com catupiry, borda recheada.

A green "Concluir pedido" (Finish order) button is visible at the bottom right of the modal.

At the bottom of the screen, there are status indicators: "Ln 23, Col 21 (22 selected)", "Spaces: 2", "UTF-8 CRLF", "TypeScript JSX", and "Prettier".

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file.
- Content:** The code displays a modal dialog for order details. It includes a button with a red icon, a title "Detalhes do pedido", a table section showing "Mesa 55" with two items: "1 - Pizza catupiry" and "3 - Pizza calabresa".

```
<button size={40} color="#FF3f4b" />
</button>

<article className={styles.container}>
  <h2>Detalhes do pedido</h2>

  <span className={styles.table}>
    Mesa <b>{order[0].order.table}</b>
  </span>

  <section className={styles.item}>
    <span>1 - <b>Pizza catupiry</b></span>
    <span className={styles.description}>Pizza de frango com catupiry, borda recheada</span>
  </section>

  <section className={styles.item}>
    <span>3 - <b>Pizza calabresa</b></span>
    <span className={styles.description}>Pizza de frango com catupiry, borda recheada</span>
  </section>

  <button className={styles.buttonOrder}>
    Concluir pedido
  </button>
</article>
```

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- Content:** A modal window titled "Detalhes do pedido" shows the order details for table 55, listing one "Pizza catupiry" and three "Pizza calabresa".
- Buttons:** A green "Concluir pedido" button is visible at the bottom right of the modal.

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file, showing TypeScript code for a modal component.
- Code:** The code displays a modal dialog with a button to close it, a section for order details, and a list of items (e.g., "1 - Pizza catupiry").

```
<button size={40} color="#FF3f4b" />
</button>

<article className={styles.container}>
  <h2>Detalhes do pedido</h2>
  <span className={styles.table}>
    Mesa <b>{order[0].order.table}</b>
  </span>
  {order.map( item => (
    <section className={styles.item}>
      <span>1 - <b>Pizza catupiry</b></span>
      <span className={styles.description}>Pizza de frango com catupiry, borda recheada</span>
    </section>
  ))}
  <button className={styles.buttonOrder}>
    Concluir pedido
  </button>
</article>
</section>
</dialog>
```

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- Content:** A modal dialog titled "Detalhes do pedido" is displayed. It shows the table number (Mesa 55) and a list of items: "1 - Pizza catupiry" (Pizza de frango com catupiry, borda recheada). A green "Concluir pedido" button is at the bottom.

frontend

order.tsx index.tsx ... \ modal index.tsx ... \ orders

rc > app > dashboard > components > modal > index.tsx > Modalorder > order.map() callback

```
16    <X size={40} color="#FF3f4b" />
17    </button>
18
19    <article className={styles.container}>
20        <h2>Detalhes do pedido</h2>
21
22        <span className={styles.table}>
23            Mesa <b>{order[0].order.table}</b>
24        </span>
25
26        {order.map( item => (
27            <section className={styles.item}>
28                <span>1 - <b>Pizza catupiry</b></span>
29                <span className={styles.description}>Pizza de frango
30                    com catupiry, borda recheada</span>
31            </section>
32        ))
33
34        <button className={styles.buttonOrder}>
35            Concluir pedido
36        </button>
37
38    </article>
39
40    </section>
41
42    </dialog>
43 }
```

Sujeito Pizza - A melhor pizzaria

localhost:300...

SujeitoPizza

Categoria Produto [→]

Últimos pedidos

Mesa 35

Mesa 55

frontend

order.tsx index.tsx ... \modal X index.tsx ... \orders

rc > app > dashboard > components > modal > index.tsx > Modalorder > order.map() callback

```
16    <x size={40} color="#FF3f4b" />
17    </button>
18
19    <article className={styles.container}>
20        <h2>Detalhes do pedido</h2>
21
22        <span className={styles.table}>
23            Mesa <b>{order[0].order.table}</b>
24        </span>
25
26        {order.map( item => (
27            <section className={styles.item}>
28                <span>{item.amount} - <b>Pizza catupiry</b></span>
29                <span className={styles.description}>Pizza de frango
30                    com catupiry, borda recheada</span>
31            </section>
32        ))
33
34        <button className={styles.buttonOrder}>
35            Concluir pedido
36        </button>
37
38    </article>
39
40    </section>
41
42    )
43 }
```

EXPLORER

FRONTEND

- > .next
- > node_modules
- > public
- src
 - app
 - dashboard
 - > category
 - > components
 - > button
 - > header
 - > modal
 - index.tsx
 - styles.module.scss
 - > orders
 - index.tsx
 - styles.module.scss
 - > product
 - layout.tsx
 - page.tsx
 - > fonts
 - > signup
 - favicon.ico
 - globals.scss
 - layout.tsx
 - page.module.scss
 - page.tsx

OUTLINE

TIMELINE

Ln 28, Col 19 (13 selected) Spaces: 2 UTF-8 CRLF TypeScript JSX Prettier

Sujeito Pizza - A melhor pizzaria

localhost:300...

SujeitoPizza

Categoria Produto [→]

Últimos pedidos

- Mesa 35
- Mesa 55

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file, showing TypeScript code for a modal component.
- Code:** The code displays a modal interface for viewing order details, including a button with a red icon, a table section for the order, and a list of items with descriptions like "Pizza catupiry".
- Status Bar:** Shows line 28, column 52, 14 selected, and file type TypeScript JSX.

Browser Preview:

- Title:** Sujeto Pizza - A melhor pizzaria
- URL:** localhost:300...
- Content:** The browser displays the application's homepage with the title "SujeitoPizza" and a section titled "Últimos pedidos" showing recent orders for "Mesa 35" and "Mesa 55".

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file, showing TypeScript code for a modal component.
- Code:** The code displays a modal dialog with a button to close it, an article section for order details, and a section for pizza items. It uses CSS-in-JS (styled-components) and includes a map function to iterate over order items.
- Bottom Status Bar:** Shows file statistics: Ln 28, Col 38 (19 selected), Spaces: 2, UTF-8, CRLF, and language settings: TypeScript JSX, Prettier.

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localhost:300...
- Content:** The page displays the "Últimos pedidos" (Last orders) section, which lists the last orders from "Mesa 35" and "Mesa 55".
- Header:** Categories like "Categoria" and "Produto" are visible.

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file.
- Content:** The code displays a modal component for viewing order details. It includes a large red "X" button, a title "Detalhes do pedido", a table section showing "Mesa 1" with an order item, and a map function that lists items with descriptions like "Pizza de frango com catupiry, borda recheada".

Browser Preview:

- Title:** "Sujeito Pizza - A melhor pizzaria"
- URL:** "localhost:3001"
- Content:** The page header is "SujeitoPizza". Below it is a section titled "Últimos pedidos" with a refresh icon. It lists two recent orders: "Mesa 35" and "Mesa 55".

Bottom Status Bar:

- Line 30, Col 15 (44 selected)
- Spaces: 2
- UTF-8
- CRLF
- TypeScript JSX
- Prettier

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file, showing TypeScript code for a modal component.
- Content:** The code displays a modal interface with a button, an article section for order details, and a list of items from the "order.map" function.

```
<X size={40} color="#FF3f4b" />
</button>

<article className={styles.container}>
  <h2>Detalhes do pedido</h2>
  <span className={styles.table}>
    Mesa <b>{order[0].order.table}</b>
  </span>
  {order.map( item => (
    <section className={styles.item}>
      <span>{item.amount} - <b>{item.product.name}</b></span>
      <span className={styles.description}>
        {item.product.description}
      </span>
    </section>
  ))}
  <button className={styles.buttonOrder}>
    Concluir pedido
  </button>
</article>
</section>
</dialog>
}
```

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localhost:300...
- Content:** Displays the "Últimos pedidos" (Last orders) section, listing "Mesa 35" and "Mesa 55".

frontend

EXPLORER

FRONTEND

- > .next
- > node_modules
- > public
- src
 - app
 - dashboard
 - > category
 - > components
 - > button
 - > header
 - > modal
 - index.tsx
 - styles.module.scss
 - orders
 - index.tsx
 - styles.module.scss
 - product
 - layout.tsx
 - page.tsx
 - fonts
 - signup
 - favicon.ico
 - globals.scss
 - layout.tsx
 - page.module.scss
 - page.tsx

OUTLINE

TIMELINE

order.tsx

index.tsx ... \modal X

index.tsx ... \orders

Ln 26, Col 21 (3 selected) Spaces: 2 UTF-8 CRLF TypeScript JSX Prettier

```
16    <x size={40} color="#FF3f4b" />
17  </button>
18
19  <article className={styles.container}>
20    <h2>Detalhes do pedido</h2>
21
22    <span className={styles.table}>
23      Mesa <b>{order[0].order.table}</b>
24    </span>
25
26    {order.map( item => (
27      <section className={styles.item}>
28        <span>{item.amount} - <b>{item.product.name}</b></span>
29        <span className={styles.description}>
30          {item.product.description}</span>
31      </section>
32    ))}
33
34    <button className={styles.buttonOrder}>
35      Concluir pedido
36    </button>
37
38  </article>
39
40</section>
41
42</dialog>
43
44}
```

Sujeito Pizza - A melhor pizzaria

localhost:300...

SujeitoPizza

Categoria Produto [→]

Últimos pedidos

- Mesa 35
- Mesa 55

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file, showing TypeScript code for a modal component.
- Search:** Search bar at the top.
- Icons:** Standard VS Code icons for file operations.

```
order.tsx
index.tsx ...\\modal X
index.tsx ...\\orders
```

```
src > app > dashboard > components > modal > index.tsx > Modalorder > order.map() callback
```

```
16    <x size={40} color="#FF3f4b" />
17    </button>

18
19    <article className={styles.container}>
20        <h2>Detalhes do pedido</h2>

21        <span className={styles.table}>
22            Mesa <b>{order[0].order.table}</b>
23        </span>

24        {order.map( item => (
25            <section className={styles.item} key={item.id}>
26                <span>{item.amount} - <b>{item.product.name}</b></span>
27                <span className={styles.description}>
28                    {item.product.description}</span>
29                </section>
30            )}

31            <button className={styles.buttonOrder}>
32                Concluir pedido
33            </button>
34
35        </article>
36
37    </section>
38
39</dialog>
40
41
42
43
44 }
```

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localhost:300...
- Content:** Displays the "Últimos pedidos" section of the application, showing two recent orders: "Mesa 35" and "Mesa 55".
- UI Elements:** Header with "CATEGORIA" and "PRODUTO" buttons, and a back/forward navigation bar.

The image shows a developer's workspace with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file.
- Content:** The code displays a modal dialog for viewing order details. It includes a close button, a header, a table for the order, and a section for individual items with their descriptions. A "Concluir pedido" button is at the bottom.

```
<button size={40} color="#FF3f4b" />
</button>

<article className={styles.container}>
  <h2>Detalhes do pedido</h2>

  <span className={styles.table}>
    Mesa <b>{order[0].order.table}</b>
  </span>

  {order.map( item => (
    <section className={styles.item} key={item.id}>
      <span>{item.amount} - <b>{item.product.name}</b></span>
      <span className={styles.description}>
        {item.product.description}
      </span>
    </section>
  ))}
  
  <button className={styles.buttonOrder}>
    Concluir pedido
  </button>
</article>

</section>
</dialog>
```

Browser Preview:

The browser window shows a modal dialog titled "Detalhes do pedido" (Order Details) with the following content:

- Mesa 55
- 2 - Coca lata
Coca-Cola lata 350ml
- 3 - Pizza de calabresa
Pizza de calabresa saborosa

A green "Concluir pedido" button is at the bottom right of the modal.

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file.
- Content:** The code displays a modal component for viewing order details. It includes a close button, a header, a table for the table number, and a section for listing items with their amounts and descriptions. A "Concluir pedido" button is also present.

```
<button size={40} color="#FF3f4b" />
</button>

<article className={styles.container}>
  <h2>Detalhes do pedido</h2>
  <span className={styles.table}>
    Mesa <b>{order[0].order.table}</b>
  </span>
  {order.map( item => (
    <section className={styles.item} key={item.id}>
      <span>{item.amount} - <b>{item.product.name}</b></span>
      <span className={styles.description}>
        {item.product.description}
      </span>
    </section>
  ))}
  <button className={styles.buttonOrder}>
    Concluir pedido
  </button>
</article>
</section>
</dialog>
}
```

Browser Preview:

The browser shows a modal dialog titled "Detalhes do pedido" (Order Details) with the following content:

- Mesa 35
- 2 - Burger Triplo Dev
Hamburger Triplo com Cheddar
- 1 - Dev Burger Salada
Burgão Dev de Salada com 1 hambúrguer de 160g

A green "Concluir pedido" button is at the bottom right of the modal.



Search.. Ctrl + P

Invite

Pizzaria ▾

Sujeito Programador ▾

Run

POST ▾ http://localhost:3333/order

Send ▾

200 OK

16.4 ms

178 B

6 Days Ago ▾

Params

Body ●Auth ●Headers 4

Scripts

Docs

Preview

Headers 8

Cookies

Tests 0 / 0

→ Mock

Console

JSON ▾

```
1 <pre>{
2   "table": 35,
3   "name": "Jorge"
4 }</pre>
```

Preview ▾

```
1 <pre>{
2   "id": "ee42b24f-cdc5-48fd-87d3-1758a06afae3",
3   "table": 35,
4   "status": false,
5   "draft": true,
6   "name": "Jorge",
7   "created_at": "2024-11-29T02:28:31.944Z",
8   "update_at": "2024-11-29T02:28:31.944Z"
9 }</pre>
```

Filter

order

PUT Finalizar Pedido (Finish Order)

GET Detalhes de um Pedido (Deta...)

GET Listar Pedidos (List Orders)

PUT Enviar Pedido (Send Order)

POST Adicionar um Item ao Pedido

DEL Remover Item

POST Criar uma ORDER - Abrir Mesa ▾

DEL Deletar uma ORDER - Pedido

product

POST Cadastrar Produto

GET Listar Produto por Categoria

master



Beautify JSON

\$.store.books[*].author



Preferences

Online

Made with ❤ by Kong

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under `FRONTEND`. The `src` folder contains `app`, `dashboard`, `components`, `header`, `modal`, `index.tsx`, `styles.module.scss`, `orders`, `index.tsx`, `styles.module.scss`, `product`, `layout.tsx`, `page.tsx`, `fonts`, `signup`, `favicon.ico`, `globals.scss`, `layout.tsx`, `page.module.scss`, `page.tsx`, `lib`, `providers`, and `order.tsx`.
- order.tsx:** The current file being edited. It defines an interface `OrderItemProps` and a type `OrderContextData`. The `OrderItemProps` interface includes properties for `id`, `amount`, `created_at`, `order_id`, `product_id`, and a nested `product` object with properties `id`, `name`, `price`, `description`, `banner`, and `category_id`. It also includes an `order` object with properties `id`, `table`, `name`, `draft`, and `status`. The `OrderContextData` type includes `isOpen`, `onRequestOpen`, and `onRequestClose`.

Browser Preview:

The browser window displays the application's order details page. The title is "Sujeito Pizza - A melhor pizzaria". The page shows a modal dialog titled "Detalhes do pedido" for "Mesa 35". The modal lists two items:

- 2 - Burger Triplo Dev**: Hamburger Triplo com Cheddar
- 1 - Dev Burger Salada**: Burgão Dev de Salada com 1 hambúrguer de 160g

A green button at the bottom right of the modal says "Concluir pedido".

At the bottom of the screen, there are status indicators: `Ln 23, Col 18 (13 selected)`, `Spaces: 2`, `UTF-8`, `CRLF`, `TypeScript JSX`, `Prettier`, and a bell icon.

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- order.tsx:** The active file contains TypeScript code defining an interface `OrderItemProps` and a type `OrderContextData`. A yellow circular icon is positioned over the `name` field in the `OrderItemProps` interface.
- index.tsx ...\\modal** and **index.tsx ...\\orders**: Other files in the current folder.

Browser Preview:

The browser window displays the application "Sujeito Pizza - A melhor pizzaria" at `localhost:300...`.

- Header:** Shows the logo "SujeitoPizza" and navigation links for "Categoria" and "Produto".
- Section:** "Últimos pedidos" (Last orders) lists two entries:
 - Mesa 35
 - Mesa 55

At the bottom of the screen, there are status icons and a footer bar with the text "Ln 23, Col 5 (20 selected) Spaces: 2 CRLF { TypeScript JSX ✅ Prettier".

The image shows a developer's workspace with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file, located at `src > app > dashboard > components > modal > index.tsx`.
- Content:** The code displays a modal component for viewing order details. It uses the Context API (`use(OrderContext)`) to access the order context. The modal contains a header, a table for the order items, and a section for each item with its amount and description. A "Concluir pedido" button is also present.
- Status Bar:** Shows the current line (Ln 27), column (Col 59), and selected text count (13 selected). It also indicates spaces (Spaces: 2), encoding (UTF-8), line endings (CRLF), and code styling (TypeScript JSX, Prettier).

Browser Preview:

- Title:** "Sujeito Pizza - A melhor pizzaria"
- URL:** "localhost:300..."
- Content:** The browser shows a modal dialog titled "Detalhes do pedido" (Order Details) for order number 35. It lists the items: "2 - Burger Triplo Dev" (Hamburger Triplo com Cheddar) and "1 - Dev Burger Salada" (Burgão Dev de Salada com 1 hambúrguer de 160g). A green "Concluir pedido" button is visible at the bottom right of the modal.

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file.
- Content:** The code is a modal component for viewing order details. It uses TypeScript and React. A yellow lightbulb icon is shown over the line `Mesa {order[0].order.table}`.

```
const { onRequestClose, order } = use(OrderContext);

return(
  <dialog className={styles.dialogContainer}>
    <section className={styles.dialogContent}>
      <button className={styles.dialogBack} onClick={onRequestClose}>
        <X size={40} color="#FF3f4b" />
      </button>

      <article className={styles.container}>
        <h2>Detalhes do pedido</h2>

        <span className={styles.table}>
          Mesa <b>{order[0].order.table}</b>
        </span>

        {order.map( item => (
          <section className={styles.item} key={item.id}>
            <span>{item.amount} - <b>{item.product.name}</b></span>
            <span className={styles.description}>
              {item.product.description}
            </span>
          </section>
        ))}
      </article>
    </section>
  </dialog>
)
```

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localhost:300...
- Content:** A modal window titled "Detalhes do pedido" is displayed. It shows a table entry for "Mesa 35" with the order "2 - Burger Triplo Dev" (Hamburger Triplo com Cheddar). Below it, another entry "1 - Dev Burger Salada" (Burgão Dev de Salada com 1 hambúrguer de 160g) is listed. A green button at the bottom right says "Concluir pedido".

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file.
- Content:** The code is a modal component for displaying order details. It includes sections for the dialog container, dialog content, and order details. A yellow warning icon is present on line 26.

```
<dialog className={styles.dialogContainer}>
  <section className={styles.dialogContent}>
    <button className={styles.dialogBack} onClick={onRequestClose}>
      <X size={40} color="#FF3f4b" />
    </button>

    <article className={styles.container}>
      <h2>Detalhes do pedido</h2>

      <span className={styles.table}>
        Mesa <b>{order[0].order.table}</b>
      </span>

      <span className={styles.table}>
        Mesa <b>{order[0].order.table}</b>
      </span>

      {order.map( item => (
        <section className={styles.item} key={item.id}>
          <span>{item.amount} - <b>{item.product.name}</b></span>
          <span className={styles.description}>
            {item.product.description}
          </span>
        </section>
      ))}

      <button className={styles.buttonOrder}>
        Concluir pedido
      </button>
    </article>
```

Browser Preview:

The browser shows a modal window titled "Detalhes do pedido" (Order Details) with the following content:

- Mesa 35 Mesa 35
- 2 - Burger Triplo Dev
Hambúrguer triplo com Cheddar
- 1 - Dev Burger Salada
Burgão Dev de Salada com 1 hambúrguer de 160g

A green "Concluir pedido" (Finish Order) button is at the bottom right of the modal.

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file.
- Content:** The code is a modal component for displaying order details. It includes sections for the table, items, and a button to "Concluir pedido".

```
<dialog className={styles.dialogContainer}>
  <section className={styles.dialogContent}>
    <button className={styles.dialogBack} onClick={onRequestClose}>
      <X size={40} color="#FF3f4b" />
    </button>

    <article className={styles.container}>
      <h2>Detalhes do pedido</h2>

      <span className={styles.table}>
        Mesa <b>{order[0].order.table}</b>
      </span>

      <span className={styles.table}>
        Mesa <b>{order[0].order.table}</b>
      </span>

      {order.map( item => (
        <section className={styles.item} key={item.id}>
          <span>{item.amount} - <b>{item.product.name}</b></span>
          <span className={styles.description}>
            {item.product.description}
          </span>
        </section>
      ))}
    </article>
  </section>
</dialog>
```

Browser Preview:

The browser window displays the application's interface for "Sujeito Pizza". A modal dialog is open, titled "Detalhes do pedido", showing the details of an order. The order consists of two items: "2 - Burger Triplo Dev" (Hamburger Triplo com Cheddar) and "1 - Dev Burger Salada" (Burgão Dev de Salada com 1 hambúrguer de 160g). A green button at the bottom right of the modal says "Concluir pedido".

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- Search:** Shows search results.
- File:** Shows file details.
- Outline:** Shows code navigation.
- Timeline:** Shows project history.
- Editor:** Displays the file `index.tsx` (highlighted in the tab bar) containing TypeScript code for a modal component. The code uses React and styled-components (`styles`).
- Bottom Status Bar:** Shows file path, line/col, and code analysis status.

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- Address:** localhost:300...
- Content:** A modal window titled "Detalhes do pedido" (Order Details) is displayed. It lists two items:
 - Mesa 35 Mesa 35**
2 - Burger Triplo Dev
Hamburger Triplo com Cheddar
 - 1 - Dev Burger Salada**
Burgão Dev de Salada com 1 hambúrguer de 160gA green button labeled "Concluir pedido" (Finish Order) is at the bottom right of the modal.

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file.
- Content:** The code displays a modal component for viewing order details. It includes sections for the table, name, and items, each with a count and product name. A "Concluir pedido" button is at the bottom.

```
<dialog className={styles.dialogContainer}>
  <section className={styles.dialogContent}>
    <button className={styles.dialogBack} onClick={onRequestClose}>
      <X size={40} color="#FF3f4b" />
    </button>

    <article className={styles.container}>
      <h2>Detalhes do pedido</h2>

      <span className={styles.table}>
        Mesa <b>{order[0].order.table}</b>
      </span>

      <span className={styles.name}>
        Mesa <b>{order[0].order.table}</b>
      </span>

      {order.map( item => (
        <section className={styles.item} key={item.id}>
          <span>{item.amount} - <b>{item.product.name}</b></span>
          <span className={styles.description}>
            {item.product.description}
          </span>
        </section>
      ))}

      <button className={styles.buttonOrder}>
        Concluir pedido
      </button>
    </article>
```

Browser Preview:

The browser shows a dark-themed interface for "Sujeito Pizza". A modal window is open, titled "Detalhes do pedido". It lists two items: "Mesa 35 Mesa 35", "2 - Burger Triplo Dev" (Hamburger Triplo com Cheddar), and "1 - Dev Burger Salada" (Burgão Dev de Salada com 1 hambúrguer de 160g). A green "Concluir pedido" button is visible at the bottom right of the modal.

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file.
- Content:** The code displays a modal component for viewing order details. It includes sections for the table and name of the order, and a list of items with their descriptions and amounts.

```
<dialog className={styles.dialogContainer}>
  <section className={styles.dialogContent}>
    <button className={styles.dialogBack} onClick={onRequestClose}>
      <X size={40} color="#FF3f4b" />
    </button>

    <article className={styles.container}>
      <h2>Detalhes do pedido</h2>

      <span className={styles.table}>
        Mesa <b>{order[0].order.table}</b>
      </span>

      <span className={styles.name}>
        Nome da Mesa <b>{order[0].order.table}</b>
      </span>

      {order.map( item => (
        <section className={styles.item} key={item.id}>
          <span>{item.amount} - <b>{item.product.name}</b></span>
          <span className={styles.description}>
            {item.product.description}
          </span>
        </section>
      ))}
    </article>
  </section>
</dialog>
```

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localhost:300...
- Content:** A modal window titled "Detalhes do pedido" is displayed. It shows the table number (Mesa 35), the order name (Burger Triplo Dev), its description (Hamburger Triplo com Cheddar), and a list of items (Dev Burger Salada). A green button at the bottom right says "Concluir pedido".

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file.
- Content:** The code displays a modal component for viewing order details. It includes sections for the table and name of the order, and a list of items with their descriptions.

```
<dialog className={styles.dialogContainer}>
  <section className={styles.dialogContent}>
    <button className={styles.dialogBack} onClick={onRequestClose}>
      <X size={40} color="#FF3f4b" />
    </button>

    <article className={styles.container}>
      <h2>Detalhes do pedido</h2>

      <span className={styles.table}>
        Mesa <b>{order[0].order.table}</b>
      </span>

      <span className={styles.name}>
        Nome da Mesa <b>{order[0].order.table}</b>
      </span>

      {order.map( item => (
        <section className={styles.item} key={item.id}>
          <span>{item.amount} - <b>{item.product.name}</b></span>
          <span className={styles.description}>
            {item.product.description}
          </span>
        </section>
      ))}
    </article>
  </section>
</dialog>
```

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localhost:300...
- Content:** A modal window titled "Detalhes do pedido" is displayed. It shows the table number (Mesa 35), the name of the table (Nome da Mesa 35), a list of items (2 - Burger Triplo Dev, 1 - Dev Burger Salada), and a "Concluir pedido" button.

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file, located at `src > app > dashboard > components > modal > index.tsx`.
- Content:** The code is a TypeScript component for a modal window. It includes imports for "Modalorder" and "Order". The component displays a dialog with a close button, a section for "Detalhes do pedido", and a table for "Mesa". It also lists the name of the table and the name of the order. A map function processes an array of items, displaying their amount and product name. A "Concluir pedido" button is at the bottom.

```
<dialog className={styles.dialogContainer}>
  <section className={styles.dialogContent}>
    <button className={styles.dialogBack} onClick={onRequestClose}>
      <X size={40} color="#FF3f4b" />
    </button>

    <article className={styles.container}>
      <h2>Detalhes do pedido</h2>

      <span className={styles.table}>
        Mesa <b>{order[0].order.table}</b>
      </span>

      <span className={styles.name}>
        Nome da Mesa <b>{order[0].order.name}</b>
      </span>

      {order.map( item => (
        <section className={styles.item} key={item.id}>
          <span>{item.amount} - <b>{item.product.name}</b></span>
          <span className={styles.description}>
            {item.product.description}
          </span>
        </section>
      ))}
    </article>
  </section>
</dialog>
```

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localhost:300...
- Content:** The browser shows a modal window titled "Detalhes do pedido". It displays the table "35" and the name "Nome da Mesa Jorge". Below this, it lists two items:
 - 2 - Burger Triplo Dev**: Hamburger Triplo com Cheddar
 - 1 - Dev Burger Salada**: Burgão Dev de Salada com 1 hambúrguer de 160gA green "Concluir pedido" button is at the bottom right of the modal.

frontend

order.tsx index.tsx ...\\modal X index.tsx ...\\orders

src > app > dashboard > components > modal > index.tsx > Modalorder

```
12     <dialog className={styles.dialogContainer}>
13         <section className={styles.dialogContent}>
14             <button className={styles.dialogBack} onClick=
15                 {onRequestClose}>
16                 <X size={40} color="#FF3f4b" />
17             </button>
18
19             <article className={styles.container}>
20                 <h2>Detalhes do pedido</h2>
21
22                 <span className={styles.table}>
23                     Mesa <b>{order[0].order.table}</b>
24                 </span>
25
26                 {order[0].order.name && (
27                     <span className={styles.name}>
28                         Nome da Mesa <b>{order[0].order.name}</b>
29                     </span>
30                 )}
31
32                 {order.map( item => (
33                     <section className={styles.item} key={item.id}>
34                         <span>{item.amount} - <b>{item.product.name}</b></span>
35                         <span className={styles.description}>
36                             {item.product.description}</span>
37                     </section>
38                 )))
39
40                 <button className={styles.buttonOrder}>
41                     Concluir pedido
42                 </button>
```

EXPLORER

FRONTEND

- > .next
- > node_modules
- > public
- src
 - app
 - dashboard
 - > category
 - > components
 - > button
 - > header
 - > modal
 - > index.tsx
 - > styles.module.scss
 - > orders
 - > index.tsx
 - > styles.module.scss
 - > product
 - > layout.tsx
 - > page.tsx
 - > fonts
 - > signup
 - > favicon.ico
 - > globals.scss
 - > layout.tsx
 - > page.module.scss
 - > page.tsx
- > OUTLINE
- > TIMELINE

Spaces: 2 UTF-8 CRLF TypeScript JSX Prettier

Sujeito Pizza - A melhor pizzaria

localhost:300...

SujeitoPizza

Categoria Produto [→]

Últimos pedidos

- Mesa 35
- Mesa 55

frontend

order.tsx index.tsx ...\\modal X index.tsx ...\\orders

src > app > dashboard > components > modal > index.tsx > Modalorder

```
12     <dialog className={styles.dialogContainer}>
13         <section className={styles.dialogContent}>
14             <button className={styles.dialogBack} onClick=
15                 {onRequestClose}>
16                 <X size={40} color="#FF3f4b" />
17             </button>
18
19             <article className={styles.container}>
20                 <h2>Detalhes do pedido</h2>
21
22                 <span className={styles.table}>
23                     Mesa <b>{order[0].order.table}</b>
24                 </span>
25
26                 {order[0].order?.name && (
27                     <span className={styles.name}>
28                         Nome da Mesa <b>{order[0].order.name}</b>
29                     </span>
30                 )}
31
32                 {order.map( item => (
33                     <section className={styles.item} key={item.id}>
34                         <span>{item.amount} - <b>{item.product.name}</b></span>
35                         <span className={styles.description}>
36                             {item.product.description}</span>
37                     </section>
38                 )));
39
40             <button className={styles.buttonOrder}>
41                 Concluir pedido
42             </button>
```

Sujeito Pizza - A melhor pizzaria

localhost:300...

SujeitoPizza

Categoria Produto [→]

Últimos pedidos ↻

- Mesa 35
- Mesa 55

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file.
- Content:** The code is a modal component for displaying order details. It includes sections for the table, customer name, and items, along with a "Concluir pedido" button.

```
<dialog className={styles.dialogContainer}>
  <section className={styles.dialogContent}>
    <button className={styles.dialogBack} onClick={onRequestClose}>
      <X size={40} color="#FF3f4b" />
    </button>

    <article className={styles.container}>
      <h2>Detalhes do pedido</h2>

      <span className={styles.table}>
        Mesa <b>{order[0].order.table}</b>
      </span>

      {order[0].order?.name && (
        <span className={styles.name}>
          Nome da Mesa <b>{order[0].order.name}</b>
        </span>
      )}

      {order.map( item => (
        <section className={styles.item} key={item.id}>
          <span>{item.amount} - <b>{item.product.name}</b></span>
          <span className={styles.description}>
            {item.product.description}
          </span>
        </section>
      ))}
    </article>
  </section>
</dialog>
```

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localhost:300...
- Content:** A modal window titled "Detalhes do pedido" displays order details for table 35. It lists two items: "2 - Burger Triplo Dev" (Hamburger Triplo com Cheddar) and "1 - Dev Burger Salada" (Burgão Dev de Salada com 1 hambúrguer de 160g). A green "Concluir pedido" button is at the bottom right.

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "styles.module.scss" is the active file, displaying SCSS code for a modal component.
- Code:**

```
src > app > dashboard > components > modal > styles.module.scss > ...  
51 .description{  
52   color: #CCC;  
53   margin-top: 4px;  
54 }  
55 .buttonOrder{  
56   background-color: var(--green-900);  
57   padding: 8px;  
58   border-radius: 4px;  
59   border: 0;  
60   font-weight: bold;  
61 }  
62 .name {  
63   margin-left: 16px;  
64 }  
65 }  
66 }  
67 }
```
- Status Bar:** Shows line 65, column 1 (38 selected), spaces: 2, UTF-8, CRLF, SCSS, Prettier, and a bell icon.

Browser Preview:

The browser window displays a modal dialog from "Sujeito Pizza - A melhor pizzaria". The modal title is "Detalhes do pedido". It lists the following items:

- Mesa 35** Nome da Mesa Jorge
- 2 - Burger Triplo Dev**
Hambúrguer triplo com Cheddar
- 1 - Dev Burger Salada**
Burgão Dev de Salada com 1 hambúrguer de 160g

A green button at the bottom right of the modal says "Concluir pedido".

The image shows a developer's workspace with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file.
- Content:** The code is for a modal component. It includes imports from "styles.module.scss", a dialog container, and a section for "Detalhes do pedido". It also displays the table number and the name of the table.

```
<dialog className={styles.dialogContainer}>
  <section className={styles.dialogContent}>
    <button className={styles.dialogBack} onClick={onRequestClose}>
      <X size={40} color="#FF3f4b" />
    </button>

    <article className={styles.container}>
      <h2>Detalhes do pedido</h2>

      <span className={styles.table}>
        Mesa <b>{order[0].order.table}</b>
      </span>

      {order[0].order?.name && (
        <span className={styles.name}>
          Nome da Mesa <b>{order[0].order.name}</b>
        </span>
      )}

      {order.map( item => (
        <section className={styles.item} key={item.id}>
          <span>{item.amount} - <b>{item.product.name}</b></span>
          <span className={styles.description}>
            {item.product.description}
          </span>
        </section>
      ))}

      <button className={styles.buttonOrder}>
        Concluir pedido
      </button>
    </article>
  </section>
</dialog>
```

Browser Preview:

The browser shows a modal window titled "Detalhes do pedido". It displays the table number (Mesa 35) and the name of the table (Nome da Mesa Jorge). Below this, it lists two items: "2 - Burger Triplo Dev" (Hamburger Triplo com Cheddar) and "1 - Dev Burger Salada" (Burgão Dev de Salada com 1 hambúrguer de 160g). A green button at the bottom right says "Concluir pedido".

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file.
- Content:** The code is for a modal component. It includes imports from "styles.module.scss", a dialog container, and a section for order details. It uses state management and map functions to render items.
- Bottom Status Bar:** Shows line 28, column 15, 28 selected, and file type TypeScript JSX.

Browser Preview:

- Title:** "Sujeito Pizza - A melhor pizzaria"
- URL:** "localhost:300..."
- Content:** A modal window titled "Detalhes do pedido" (Order Details) is displayed. It shows an order for "Mesa 35 Jorge" with two items:
 - 2 - Burger Triplo Dev: Hamburger Triplo com Cheddar
 - 1 - Dev Burger Salada: Burgão Dev de Salada com 1 hambúrguer de 160gA green "Concluir pedido" (Finish Order) button is at the bottom.

frontend

order.tsx index.tsx ...\\modal X styles.module.scss index.tsx ...\\ ...

src > app > dashboard > components > modal > index.tsx > Modalorder

```
12     <dialog className={styles.dialogContainer}>
13         <section className={styles.dialogContent}>
14             <button className={styles.dialogBack} onClick={onRequestClose}>
15                 <X size={40} color="#FF3f4b" />
16             </button>
17
18             <article className={styles.container}>
19                 <h2>Detalhes do pedido</h2>
20
21                 <span className={styles.table}>
22                     Mesa <b>{order[0].order.table}</b>
23                 </span>
24
25
26                 {order[0].order?.name && (
27                     <span className={styles.name}>
28                         <b>{order[0].order.name}</b>
29                     </span>
30                 )}
31
32                 {order.map( item => (
33                     <section className={styles.item} key={item.id}>
34                         <span>{item.amount} - <b>{item.product.name}</b></span>
35                         <span className={styles.description}>
36                             {item.product.description}</span>
37                         </section>
38                     )));
39
40             <button className={styles.buttonOrder}>
41                 Concluir pedido
42             </button>
```

Ln 31, Col 1 Spaces: 2 UTF-8 CRLF { TypeScript JSX ✅ Prettier

EXPLORER FRONTEND OUTLINE TIMELINE

Sujeito Pizza - A melhor pizzaria

localhost:300...

SujeitoPizza

Categoria Produto [→]

Últimos pedidos ⌂

- Mesa 35
- Mesa 55

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file.
- Content:** The code is a modal component for displaying order details. It includes imports for "order", "styles.module.scss", and "index.tsx". The component uses a dialog to show the order details, including the table number and items ordered.

```
<dialog className={styles.dialogContainer}>
  <section className={styles.dialogContent}>
    <button className={styles.dialogBack} onClick={onRequestClose}>
      <X size={40} color="#FF3f4b" />
    </button>

    <article className={styles.container}>
      <h2>Detalhes do pedido</h2>

      <span className={styles.table}>
        Mesa <b>{order[0].order.table}</b>
      </span>

      {order[0].order?.name && (
        <span className={styles.name}>
          <b>{order[0].order.name}</b>
        </span>
      )}

      {order.map( item => (
        <section className={styles.item} key={item.id}>
          <span>{item.amount} - <b>{item.product.name}</b></span>
          <span className={styles.description}>
            {item.product.description}
          </span>
        </section>
      ))}
    </article>
  </section>
</dialog>
```

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localhost:300...
- Content:** A modal window titled "Detalhes do pedido" displays the order details for table 55. It lists 2 Coca Lata (Coca-Cola lata 350ml) and 3 Pizza de calabresa (Pizza de calabresa saborosa). A green button at the bottom right says "Concluir pedido".

frontend

EXPLORER

FRONTEND

- > .next
- > node_modules
- > public
- src
 - app
 - dashboard
 - > category
 - > components
 - > button
 - > header
 - > modal
 - index.tsx
 - styles.module.scss
 - orders
 - index.tsx
 - styles.module.scss
 - product
 - layout.tsx
 - page.tsx
 - fonts
 - signup
 - favicon.ico
 - globals.scss
 - layout.tsx
 - page.module.scss
 - page.tsx

OUTLINE

TIMELINE

order.tsx

index.tsx ... \ modal

styles.module.scss

index.tsx ... \ ...

```
src > app > dashboard > components > modal > index.tsx > Modalorder
12     <dialog className={styles.dialogContainer}>
13         <section className={styles.dialogContent}>
14             <button className={styles.dialogBack} onClick={onRequestClose}>
15                 <X size={40} color="#FF3f4b" />
16             </button>
17
18             <article className={styles.container}>
19                 <h2>Detalhes do pedido</h2>
20
21                 <span className={styles.table}>
22                     Mesa <b>{order[0].order.table}</b>
23                 </span>
24
25
26                 {order[0].order?.name && (
27                     <span className={styles.name}>
28                         <b>{order[0].order.name}</b>
29                     </span>
30                 )}
31
32
33                 {order.map( item => (
34                     <section className={styles.item} key={item.id}>
35                         <span>{item.amount} - <b>{item.product.name}</b></span>
36                         <span className={styles.description}>
37                             {item.product.description}</span>
38                     </section>
39                 )))
40
41
42             <button className={styles.buttonOrder}>
43                 Concluir pedido
44             </button>
```

Ln 31, Col 1 Spaces: 2 UTF-8 CRLF ⓘ TypeScript JSX ✅ Prettier

Sujeito Pizza - A melhor pizzaria

localhost:300...

SujeitoPizza

Categoria Produto [→]

Últimos pedidos

- Mesa 35
- Mesa 55

The screenshot shows a developer's environment with a code editor and a browser preview window.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- order.tsx:** The current file is open in the editor, displaying React component code. It includes imports from "react", "useEffect", and "useRef". The component uses "useState" and "useEffect" hooks, and includes "return" statements with JSX.
- index.tsx:** Another file is visible in the background.

Browser Preview:

- Title Bar:** "Sujeito Pizza - A melhor pizzaria"
- Address Bar:** "localhost:3000/dashboard"
- Content:** Displays the "Dashboard" page of the Sujeito Pizza application. The page features a header with the Sujeito Pizza logo, a sidebar with navigation links like "Dashboard", "Pedidos", "Produtos", and "Relatórios", and a main area with a modal dialog.
- Modal Dialog:** Titled "Detalhes do pedido", it shows information for an order at "Mesa 35" for "Jorge".
 - Order 1:** "2 - Burger Triplo Dev" (Hamburger Triplo com Cheddar)
 - Order 2:** "1 - Dev Burger Salada" (Burgão Dev de Salada com 1 hambúrguer de 160g)A green "Concluir pedido" button is at the bottom of the modal.

Bottom Status Bar: Shows file paths, line numbers (Ln 3), and other status indicators.

File Edit View Ultimate Tools Help localhost:5432/pizzaria - Beekeeper Studio

pizzaria C + Query #1 products [all] orders [all] x +

Filter ENTITIES 217 +

public

- _prisma_migrations
- categories
- items
- orders
- products
- users

information_schema

pg_catalog

| | id | table | status | draft | name | created_at... | update_at |
|---|---------------------------------------|-------|--------|-------|--------|----------------|--------------------------|
| 1 | 0d0fef8f-9947-4e0a-94d2-dc2722b9cee0 | 55 | false | false | (NULL) | 2024-11-26 ... | 2024-11-26 01:30:26.0... |
| 2 | c9019c83-9a02-4152-a475-21f07e1e39... | 9 | true | false | (NULL) | 2024-11-26 ... | 2024-11-26 15:36:31.8... |
| 3 | ee42b24f-cdc5-48fd-87d3-1758a06afae3 | 35 | false | false | Jorge | 2024-11-29 ... | 2024-11-29 02:28:31.9... |

File Edit View Ultimate Tools Help localhost:5432/pizzaria - Beekeeper Studio

pizzaria C + Query #1 products [all] orders [all] x +

Filter ENTITIES 217 +

public

- _prisma_migrations
- categories
- items
- orders
- products
- users

information_schema

pg_catalog

id equals Enter Value

| | id | table | status | draft | name | created_at... | update_at |
|---|---------------------------------------|-------|--------|-------|--------|----------------|--------------------------|
| 1 | 0d0fef8f-9947-4e0a-94d2-dc2722b9cee0 | 55 | false | false | (NULL) | 2024-11-26 ... | 2024-11-26 01:30:26.0... |
| 2 | c9019c83-9a02-4152-a475-21f07e1e39... | 9 | true | false | (NULL) | 2024-11-26 ... | 2024-11-26 15:36:31.8... |
| 3 | ee42b24f-cdc5-48fd-87d3-1758a06afae3 | 35 | false | false | Jorge | 2024-11-29 ... | 2024-11-29 02:28:31.9... |

File Edit View Ultimate Tools Help localhost:5432/pizzaria - Beekeeper Studio

pizzaria C +

Query #1 products [all] orders [all]

Filter

ENTITIES 217 +

public

- _prisma_migrations
- categories
- items
- orders
- products
- users

information_schema

pg_catalog

id table status draft name created_at update_at

| | id | table | status | draft | name | created_at | update_at |
|---|---------------------------------------|-------|--------|-------|---------|----------------|--------------------------|
| 1 | 0d0fef8f-9947-4e0a-94d2-dc2722b9cee0 | 55 | false | false | (NULL) | 2024-11-26 ... | 2024-11-26 01:30:26.0... |
| 2 | c9019c83-9a02-4152-a475-21f07e1e39... | 9 | true | false | (NULL) | 2024-11-26 ... | 2024-11-26 15:36:31.8... |
| 3 | ee42b24f-cdc5-48fd-87d3-1758a06afae3 | 35 | false | false | (EMPTY) | 2024-11-29 ... | 2024-11-29 02:28:31.9... |

pizzaria

Filter

ENTITIES 217

- public
 - _prisma_migrations
 - categories
 - items
 - orders
 - products
 - users
- information_schema
- pg_catalog

Query #1 products [all] orders [all]

id equals Enter Value

| | id | table | status | draft | name | created_at... | update_at |
|---|---------------------------------------|-------|--------|-------|---------|----------------|--------------------------|
| 1 | Od0fef8f-9947-4e0a-94d2-dc2722b9cee0 | 55 | false | false | (NULL) | 2024-11-26 ... | 2024-11-26 01:30:26.0... |
| 2 | c9019c83-9a02-4152-a475-21f07e1e39... | 9 | true | false | (NULL) | 2024-11-26 ... | 2024-11-26 15:36:31.8... |
| 3 | ee42b24f-cdc5-48fd-87d3-1758a06afae3 | 35 | false | false | (EMPTY) | 2024-11-29 ... | 2024-11-29 02:28:31.9... |

localhost:5432/pizzaria

postgresql

Structure ↗ Unknown 2 minutes ago

1 >

C + ⚙

The screenshot shows a developer's environment with a code editor and a browser preview side-by-side.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- order.tsx:** The current file is open in the editor, displaying React component code. It includes imports from "react", "useEffect", and "useRef". The component uses "useState" and "useContext" hooks, and includes "return" statements with JSX.
- index.tsx:** Another file is visible in the background.

Browser Preview:

- Title Bar:** "Sujeito Pizza - A melhor pizzaria"
- Address Bar:** "localhost:3000/dashboard"
- Content:** A modal dialog titled "Detalhes do pedido" (Order Details) is displayed. It shows an order for "Mesa 35" (Table 35) and "Jorge".
 - Items:**
 - 2 - Burger Triplo Dev**: Hamburger Triplo com Cheddar
 - 1 - Dev Burger Salada**: Burgão Dev de Salada com 1 hambúrguer de 160g
 - Buttons:** A green "Concluir pedido" (Finish order) button.

Sujeito Pizza - A melhor pizzaria

localhost:3000/dashboard

Categoria Produto []

SujeitoPizza

Últimos pedidos

Mesa 35

Mesa 55

```
order.tsx
index.tsx
src > app > dashboard > components > modal > index.tsx
  12   <dialog class="modal">
  13     <section class="modal-content">
  14       <button class="close">&times;
  15       <onRequestClose={close}>
  16         <x size={size}>
  17       </button>
  18
  19       <article>
  20         <h2>Detalhes do pedido</h2>
  21         <span>Mesa:</span>
  22         <span>Mesa 35</span>
  23
  24         <order>
  25           <span>Quantidade:</span>
  26           <span>1</span>
  27           <b>Preço:</b>
  28           <span>R$ 10,00</span>
  29         </order>
  30       </article>
  31
  32       <order>
  33         <span>Quantidade:</span>
  34         <span>1</span>
  35         <b>Preço:</b>
  36         <span>R$ 10,00</span>
  37       </order>
  38     </div>
  39
  40     <button>Concluído</button>
  41   </div>
  42 </dialog>
```

A screenshot of a developer's workspace, likely using a monolithic framework like Next.js. The interface includes:

- EXPLORER** sidebar showing the project structure, including **FRONTEND**, **.next**, **node_modules**, **public**, and **src** (expanded) containing **app** (expanded) with **dashboard**, **category**, **components** (expanded) with **button**, **header**, and **modal** (expanded) with **index.tsx** (selected), **styles.module.scss**, and **orders** (expanded) with **index.tsx** and **styles.module.scss**.
- CODE** editor showing **order.tsx** file content:

```
src > app > dashboard > components > modal > index.tsx
12   <dialog class="fixed z-50 w-1/2 h-1/2 bg-white rounded-lg overflow-hidden">
13     <div class="flex flex-col justify-center items-center">
14       <section class="w-full p-6">
15         <button type="button" class="absolute top-0 right-0 -mt-4 -mr-4" aria-label="Close">
16           <X size="24" class="text-red-500"/>
17         </button>
18         <article>
19           <h2>Detalhes do pedido
20           <span>Mesa
21           <span>35
22           <table border="1">
23             <thead>
24               <tr>
25                 <th>Item
26                 <th>Descrição
27               </tr>
28             </thead>
29             <tbody>
30               <tr>
31                 <td>2
32                 <td>2 - Burger Triplo Dev
33               </tr>
34               <tr>
35                 <td>1
36                 <td>1 - Dev Burger Salada
37               </tr>
38             </tbody>
39           </table>
40           <button type="button" class="w-full p-2 mt-4 bg-green-500 text-white rounded-lg">Concluir pedido
41         </div>
42       </div>
43     </div>
44   </dialog>
```

- BROWSER PREVIEW** showing the application running at **localhost:3000/dashboard**. The page displays a header with the brand name "Sujeito Pizza". Below it, there are sections for "Categorias", "Produtos", and "Pedidos". A modal dialog is open, titled "Detalhes do pedido" (Order Details), showing the order details for "Mesa 35":
 - 2 - Burger Triplo Dev (Hamburger Triplo com Cheddar)
 - 1 - Dev Burger Salada (Burgão Dev de Salada com 1 hambúrguer de 160g)A green button labeled "Concluir pedido" (Finish Order) is visible at the bottom of the modal.
- STATUS BAR** at the bottom left showing file counts: **0** files, **0** changes, and **0** conflicts.

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file.
- Content:** The code is a modal component for displaying order details. It includes a close button, a title "Detalhes do pedido", a table for the table number, and a map function to list items with their names and descriptions. A "Concluir pedido" button is at the bottom.

```
<dialog className={styles.dialogContainer}>
  <section className={styles.dialogContent}>
    <button className={styles.dialogBack} onClick={onRequestClose}>
      <X size={40} color="#FF3f4b" />
    </button>

    <article className={styles.container}>
      <h2>Detalhes do pedido</h2>

      <span className={styles.table}>
        Mesa <b>{order[0].order.table}</b>
      </span>

      {order[0].order?.name && (
        <span className={styles.name}>
          <b>{order[0].order.name}</b>
        </span>
      )}

      {order.map( item => (
        <section className={styles.item} key={item.id}>
          <span>{item.amount} - <b>{item.product.name}</b></span>
          <span className={styles.description}>
            {item.product.description}
          </span>
        </section>
      ))}

      <button className={styles.buttonOrder}>
        Concluir pedido
      </button>
    </article>
  </section>
</dialog>
```

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- Content:** A modal window titled "Detalhes do pedido" is displayed. It shows the table number "Mesa 35" and two items:
 - 2 - Burger Triplo Dev: Hamburger Triplo com Cheddar
 - 1 - Dev Burger Salada: Burgão Dev de Salada com 1 hambúrguer de 160gA green "Concluir pedido" button is at the bottom right of the modal.

UC: Projetos

Assunto:

Projeto Pizzaria – Frontend_02 (NextJS)

Sistema Pizzaria Web

62 – Concluindo Pedido

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file.
- Content:** The code is for a modal component. It includes imports from "styles.module.scss", a class definition for "Modalorder", and a function that returns JSX for a dialog, section, button, article, and table components. It also includes logic for displaying the order table and items based on the order status.
- Bottom Status Bar:** Shows line 30, column 13, spaces: 2, CRLF, TypeScript JSX, Prettier, and a refresh icon.

Browser Preview:

- Title:** "Sujeito Pizza - A melhor pizzaria"
- URL:** "localhost:300..."
- Content:** A modal window titled "Detalhes do pedido" (Details of the order) is displayed. It shows the table "Mesa 35" and two items:
 - 2 - Burger Triplo Dev: Hamburger Triplo com Cheddar
 - 1 - Dev Burger Salada: Burgão Dev de Salada com 1 hambúrguer de 160gA green "Concluir pedido" (Finish order) button is at the bottom right of the modal.

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file, located at `src > app > dashboard > components > modal > index.tsx`.
- Content:** The code is a TypeScript component named "Modalorder". It uses CSS-in-JS (styled-components) and includes logic to handle an order array and map over items to display product details.

```
<dialog className={styles.dialogContainer}>
  <section className={styles.dialogContent}>
    <button className={styles.dialogBack} onClick={onRequestClose}>
      <X size={40} color="#FF3f4b" />
    </button>

    <article className={styles.container}>
      <h2>Detalhes do pedido</h2>

      <span className={styles.table}>
        Mesa <b>{order[0].order.table}</b>
      </span>

      {order[0].order?.name && (
        <span className={styles.name}>
          <b>{order[0].order.name}</b>
        </span>
      )}

      {order.map( item => (
        <section className={styles.item} key={item.id}>
          <span>{item.amount} - <b>{item.product.name}</b></span>
          <span className={styles.description}>
            {item.product.description}
          </span>
        </section>
      ))}

      <button className={styles.buttonOrder}>
        Concluir pedido
      </button>
    </article>
  </section>
</dialog>
```

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localhost:300...
- Content:** Displays the "Últimos pedidos" (Last orders) section, showing two entries: "Mesa 35" and "Mesa 55".

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file, showing TypeScript code for a modal component.
- Content:** The code defines a modal dialog with a close button, a header "Detalhes do pedido", and a table section displaying order details. It also includes logic to map over the order items.
- Bottom Status Bar:** Shows line 38, column 14, spaces: 2, encoding: UTF-8, CRLF, and Prettier status.

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localhost:300...
- Content:** Displays a modal window titled "Detalhes do pedido" with the following details:
 - Mesa 55
 - 2 - Coca lata: Coca-Cola lata 350ml
 - 3 - Pizza de calabresa: Pizza de calabresa saborosaA green "Concluir pedido" button is at the bottom right of the modal.

frontend

order.tsx index.tsx ...\\modal styles.module.scss index.tsx ...\\...

src > app > dashboard > components > modal > index.tsx > Modalorder

```
12     <dialog className={styles.dialogContainer}>
13         <section className={styles.dialogContent}>
14             <button className={styles.dialogBack} onClick=
15                 {onRequestClose}>
16                 <X size={40} color="#FF3f4b" />
17             </button>
18
19             <article className={styles.container}>
20                 <h2>Detalhes do pedido</h2>
21
22                 <span className={styles.table}>
23                     Mesa <b>{order[0].order.table}</b>
24                 </span>
25
26                 {order[0].order?.name && (
27                     <span className={styles.name}>
28                         <b>{order[0].order.name}</b>
29                     </span>
30                 )}
31
32                 {order.map( item => (
33                     <section className={styles.item} key={item.id}>
34                         <span>{item.amount} - <b>{item.product.name}</b></span>
35                         <span className={styles.description}>
36                             {item.product.description}</span>
37                         </section>
38                 ))}
39
40             <button className={styles.buttonOrder}>
41                 Concluir pedido
42             </button>
```

Ln 38, Col 14 Spaces: 2 UTF-8 CRLF TypeScript JSX Prettier

Sujeito Pizza - A melhor pizzaria

localhost:300...

SujeitoPizza

Categoria Produto [→]

Últimos pedidos

Mesa 35

Mesa 55

The image shows a developer's workspace with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file, located at `src > app > dashboard > components > modal > index.tsx`. The code is a modal component for displaying order details.
- Content:** The code uses TypeScript and JSX. It includes imports for `Modalorder` and `styles.module.scss`. The component structure is as follows:

```
<dialog className={styles.dialogContainer}>
  <section className={styles.dialogContent}>
    <button className={styles.dialogBack} onClick={onRequestClose}>
      <X size={40} color="#FF3f4b" />
    </button>

    <article className={styles.container}>
      <h2>Detalhes do pedido</h2>

      <span className={styles.table}>
        Mesa <b>{order[0].order.table}</b>
      </span>

      {order[0].order?.name && (
        <span className={styles.name}>
          <b>{order[0].order.name}</b>
        </span>
      )}

      {order.map( item => (
        <section className={styles.item} key={item.id}>
          <span>{item.amount} - <b>{item.product.name}</b></span>
          <span className={styles.description}>
            {item.product.description}
          </span>
        </section>
      ))}
    </article>
  </section>
</dialog>
```

- Status Bar:** Shows file statistics: Ln 38, Col 14, Spaces: 2, UTF-8, CRLF, and Prettier status.

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localhost:300...
- Content:** A modal window titled "Detalhes do pedido" is displayed. It shows the order details for table 55:
 - Mesa 55
 - 2 - Coca lata: Coca-Cola lata 350ml
 - 3 - Pizza de calabresa: Pizza de calabresa saborosaA green button labeled "Concluir pedido" is at the bottom right of the modal.

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file.
- Content:** The code displays a modal component for order details. It includes logic to check if an order has a name and maps over an array of items to display their descriptions and amounts.
- Line 40:** A yellow circular icon with a question mark is positioned next to the button element, indicating a potential issue or warning.

```
<span className={styles.table}>
  Mesa <b>{order[0].order.table}</b>
</span>

{order[0].order?.name && (
  <span className={styles.name}>
    <b>{order[0].order.name}</b>
  </span>
)}

{order.map( item => (
  <section className={styles.item} key={item.id}>
    <span>{item.amount} - <b>{item.product.name}</b></span>
    <span className={styles.description}>
      {item.product.description}
    </span>
  </section>
))}

<button className={styles.buttonOrder}>
  Concluir pedido
</button>
```

Browser Preview:

The browser window shows a modal dialog titled "Detalhes do pedido" (Order Details) from "Sujeito Pizza". The modal displays the following information:

- Mesa 55
- 2 - Coca lata
Coca-Cola lata 350ml
- 3 - Pizza de calabresa
Pizza de calabresa saborosa

A green "Concluir pedido" (Finish order) button is at the bottom right of the modal.

frontend

order.tsx index.tsx ...\\modal 1 styles.module.scss index.tsx ...

src > app > dashboard > components > modal > index.tsx > Modalorder

```
21
22     <span className={styles.table}>
23         Mesa <b>{order[0].order.table}</b>
24     </span>
25
26     {order[0].order?.name && (
27         <span className={styles.name}>
28             <b>{order[0].order.name}</b>
29         </span>
30     )}
31
32     {order.map( item => (
33         <section className={styles.item} key={item.id}>
34             <span>{item.amount} - <b>{item.product.name}</b></span>
35             <span className={styles.description}>
36                 {item.product.description}</span>
37             </section>
38         )));
39
40         <button className={styles.buttonOrder} onClick={handleFinishOrder}>
41             Concluir pedido
42         </button>
43
44     </article>
45
46     </section>
47
48     </dialog>
49
50 }
```

Ln 40, Col 50 (27 selected) Spaces: 2 CRLF TypeScript JSX Prettier

Sujeito Pizza - A melhor pizzaria

localhost:300...

SujeitoPizza

Categoria Produto [→]

Últimos pedidos ↻

- Mesa 35
- Mesa 55

```
order.tsx index.tsx ...\\modal styles.module.scss index.tsx ...\\...  
src > app > dashboard > components > modal > index.tsx > Modalorder  
1 "use client"  
2  
3 import styles from './styles.module.scss'  
4 import { Import, X } from 'lucide-react'  
5 import { use } from 'react'  
6 import { OrderContext } from '@/providers/order'  
7  
8 export function Modalorder(){  
9   const { onRequestClose, order } = use(OrderContext);  
10  
11   async function handleFinishOrder() {  
12     alert("TESTE")  
13   }  
14  
15   return(  
16     <dialog className={styles.dialogContainer}>  
17       <section className={styles.dialogContent}>  
18         <button className={styles.dialogBack} onClick={onRequestClose}>  
19           <X size={40} color="#FF3f4b" />  
20         </button>  
21  
22         <article className={styles.container}>  
23           <h2>Detalhes do pedido</h2>  
24  
25           <span className={styles.table}>  
26             Mesa <b>{order[0].order.table}</b>  
27           </span>  
28  
29           {order[0].order?.name && (  
30             <span className={styles.name}>  
31               . . .
```

Ln 11, Col 1 (63 selected) Spaces: 2 CRLF ⓘ TypeScript JSX ✓ Prettier

SujeitoPizza

Últimos pedidos

- Mesa 35
- Mesa 55

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file.
- Content:** The code defines a modal component named "Modalorder". It uses TypeScript and React. It imports styles from "styles.module.scss" and the "lucide-react" library. It uses the "use" hook from "react" and the "OrderContext" provider from "@/providers/order".

```
1  "use client"
2
3  import styles from './styles.module.scss'
4  import { Import, X } from 'lucide-react'
5  import { use } from 'react'
6  import { OrderContext } from '@/providers/order'
7
8  export function Modalorder(){
9    const { onRequestClose, order } = use(OrderContext);
10
11   async function handleFinishOrder() {
12     alert("TESTE")
13   }
14
15   return(
16     <dialog className={styles.dialogContainer}>
17       <section className={styles.dialogContent}>
18         <button className={styles.dialogBack} onClick={onRequestClose}>
19           <X size={40} color="#FF3f4b" />
20         </button>
21
22         <article className={styles.container}>
23           <h2>Detalhes do pedido</h2>
24
25           <span className={styles.table}>
26             Mesa <b>{order[0].order.table}</b>
27           </span>
28
29           {order[0].order?.name && (
30             <span className={styles.name}>
31               ...
```

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localhost:300...
- Content:** A dark-themed modal window titled "Detalhes do pedido" is displayed. It shows a table with one row: "Mesa 55" with "2 - Coca lata" and "Coca-Cola lata 350ml". Below it, another row is listed: "3 - Pizza de calabresa" with "Pizza de calabresa saborosa". A green button at the bottom right says "Concluir pedido".

Bottom Status Bar:

- Ln 12, Col 19
- Spaces: 2
- UTF-8
- CRLF
- TypeScript JSX
- Prettier

The image shows a developer's environment with multiple windows open:

- VS Code (Left):** The Explorer sidebar shows the project structure under "FRONTEND". The "modal" folder is expanded, showing files like "index.tsx", "styles.module.scss", and "order.tsx". The "index.tsx" file is currently selected and displayed in the main editor area.
- Browser Preview (Top Right):** A browser window displays the URL "localhost:3000". The page content includes "localhost:3000 diz" and "TESTE". There is also a modal dialog titled "Detalhes do pedido" with details about a meal and a green "Concluir pedido" button.
- Application Interface (Bottom Right):** A dark-themed application window titled "Sujeito Pizza - A melhor pizzaria" shows a meal summary:
 - Mesa 55
 - 2 - Coca lata
Coca-Cola lata 350ml
 - 3 - Pizza de calabresa
Pizza de calabresa saborosaA green "Concluir pedido" button is at the bottom right of the modal.

Code Editor Content (index.tsx):

```
order.tsx index.tsx ...\\modal X styles.module.scss index.tsx ...\\ ...
src > app > dashboard > components > modal > index.tsx > Modalorder > handleFinishOrder
1  "use client"
2
3  import styles from './styles.module.scss'
4  import { Import, X } from 'lucide-react'
5  import { use } from 'react'
6  import { OrderContext } from '@/providers/order'
7
8  export function Modalorder(){
9    const { onRequestClose, order } = use(OrderContext);
10
11   async function handleFinishOrder() {
12     alert("TESTE")
13   }
14
15   return(
16     <dialog className={styles.dialogContainer}>
17       <section className={styles.dialogContent}>
18         <button className={styles.dialogBack} onClick={onRequestClose}>
19           <X size={40} color="#FF3f4b" />
20         </button>
21
22         <article className={styles.container}>
23           <h2>Detalhes do pedido</h2>
24
25           <span className={styles.table}>
26             Mesa <b>{order[0].order.table}</b>
27           </span>
28
29           {order[0].order?.name && (
30             <span className={styles.name}>
31               ...
32             ...
33           </span>
34         )
35       )
36     </dialog>
37   )
38 }
```

Page Footer: Ln 12, Col 19 Spaces: 2 UTF-8 CRLF { TypeScript JSX Prettier

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file.
- Content:** The code defines a modal component named "Modalorder". It imports styles from "styles.module.scss", uses the "useClient" hook, and the "OrderContext" provider. It contains an "async function handleFinishOrder()" which alerts "TESTE".
- Bottom Status Bar:** Shows "Ln 12, Col 19" and "TypeScript JSX".

Browser Preview:

- Title:** "Sujeito Pizza - A melhor pizzaria"
- URL:** "localhost:300..."
- Content:** A modal window titled "Detalhes do pedido" (Details of the order) is displayed. It shows the order details for "Mesa 55":
 - 2 - Coca lata (Coca-Cola lata 350ml)
 - 3 - Pizza de calabresa (Pizza de calabresa saborosa)A green button labeled "Concluir pedido" (Finish order) is at the bottom right.

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (Left):

- Explorer:** Shows the project structure under "FRONTEND".
- order.tsx:** The active file contains TypeScript code for handling orders. It includes functions for setting the order and opening a modal, and an asynchronous function to finish an order using an API call and a token.
- index.tsx ...\\modal:** A file related to the modal component.
- styles.module.scss:** A CSS module file.
- index.tsx ...\\...**: Another file related to the index.

```
60
61     setOrder(response.data);
62     setIsOpen(true);
63 }
64
65 function onRequestClose(){
66     setIsOpen(false);
67 }
68
69 async function finishOrder(order_id: string) {
70     const token = getCookieClient();
71
72     const data = {
73         order_id: order_id,
74     }
75
76     try {
77         await api.put("/order/finish", data, {
78             headers: {
79                 Authorization: `Bearer ${token}`
80             }
81         })
82     } catch(err){
83         console.log(err);
84         toast.error("Falha ao finalizar este pedido!");
85         return;
86     }
87
88     toast.success("Pedido finalizado com sucesso!");
89
90 }
91
92 return
```

Browser Preview (Right):

The browser shows the application "SujeitoPizza - A melhor pizzaria" at [localhost:300...](http://localhost:3000). The page displays the title "SujeitoPizza" and a section titled "Últimos pedidos" (Last orders) showing two entries: "Mesa 35" and "Mesa 55".

frontend

EXPLORER

FRONTEND

- src
 - providers
 - order.tsx
 - index.tsx ...\\modal
 - styles.module.scss
 - index.tsx ...\\...

src > providers > order.tsx > OrderContextData

```
13 product:{  
14   id: string;  
15   name: string;  
16   price: string;  
17   description: string;  
18   banner: string;  
19   category_id: string;  
20 };  
21 order:{  
22   id: string;  
23   table: number;  
24   name: string | null;  
25   draft: boolean;  
26   status: boolean;  
27 }  
28 }  
29  
30 type OrderContextData = {  
31   isOpen: boolean;  
32   onRequestOpen: (order_id: string) => Promise<void>;  
33   onRequestClose: () => void;  
34   order: OrderItemProps[];  
35   finishOrder: (order_id: string) => Promise<void>;  
36 }  
37  
38 type OrderProviderProps = {  
39   children: ReactNode;  
40 }  
41  
42 export const OrderContext = createContext({} as OrderContextData)  
43  
44 export function OrderProvider({ children }: OrderProviderProps){
```

Ln 35, Col 52 (49 selected) Spaces: 2 UTF-8 CRLF TypeScript JSX Prettier

Sujeito Pizza - A melhor pizzaria

localhost:300...

SujeitoPizza

Categoria Produto [→]

Últimos pedidos

- Mesa 35
- Mesa 55

The image shows a developer's environment split between a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "order.tsx" is the active file.
- Content:** The code handles an error during order completion and displays a success toast message. It uses the `OrderContext.Provider` component to provide context to its children.

```
81     }
82     })
83   }catch(err){
84     console.log(err);
85     toast.error("Falha ao finalizar este pedido!")
86     return;
87   }
88
89   toast.success("Pedido finalizado com sucesso!")
90
91 }
92
93 return(
94   <OrderContext.Provider
95     value={{
96       isOpen,
97       onRequestOpen,
98       onRequestClose,
99       finishOrder,
100      order
101    }}
102   >
103   {children}
104 </OrderContext.Provider>
105 )
106 }
```

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localhost:300...
- Content:** Displays the "Últimos pedidos" section, listing "Mesa 35" and "Mesa 55".

frontend

order.tsx index.tsx ...\\modal styles.module.scss index.tsx ...\\... finishOrder

```
1 "use client"
2
3 import styles from './styles.module.scss'
4 import { Import, X } from 'lucide-react'
5 import { use } from 'react'
6 import { OrderContext } from '@/providers/order'
7
8 export function Modalorder(){
9   const { onRequestClose, order, finishOrder } = use(OrderContext);
10
11   async function handleFinishOrder() {
12     alert("TESTE")
13   }
14
15   return(
16     <dialog className={styles.dialogContainer}>
17
18       <section className={styles.dialogContent}>
19         <button className={styles.dialogBack} onClick={onRequestClose}>
20           <X size={40} color="#FF3f4b" />
21         </button>
22
23         <article className={styles.container}>
24           <h2>Detalhes do pedido</h2>
25
26           <span className={styles.table}>
27             Mesa <b>{order[0].order.table}</b>
28           </span>
29
30           {order[0].order?.name && (
31             <span className={styles.name}>
```

Ln 9, Col 45 (13 selected) Spaces: 2 UTF-8 CRLF ⓘ TypeScript JSX ✓ Prettier

Sujeito Pizza - A melhor pizzaria

localhost:300...

SujeitoPizza

Categoria Produto [→]

Últimos pedidos

Mesa 35

Mesa 55

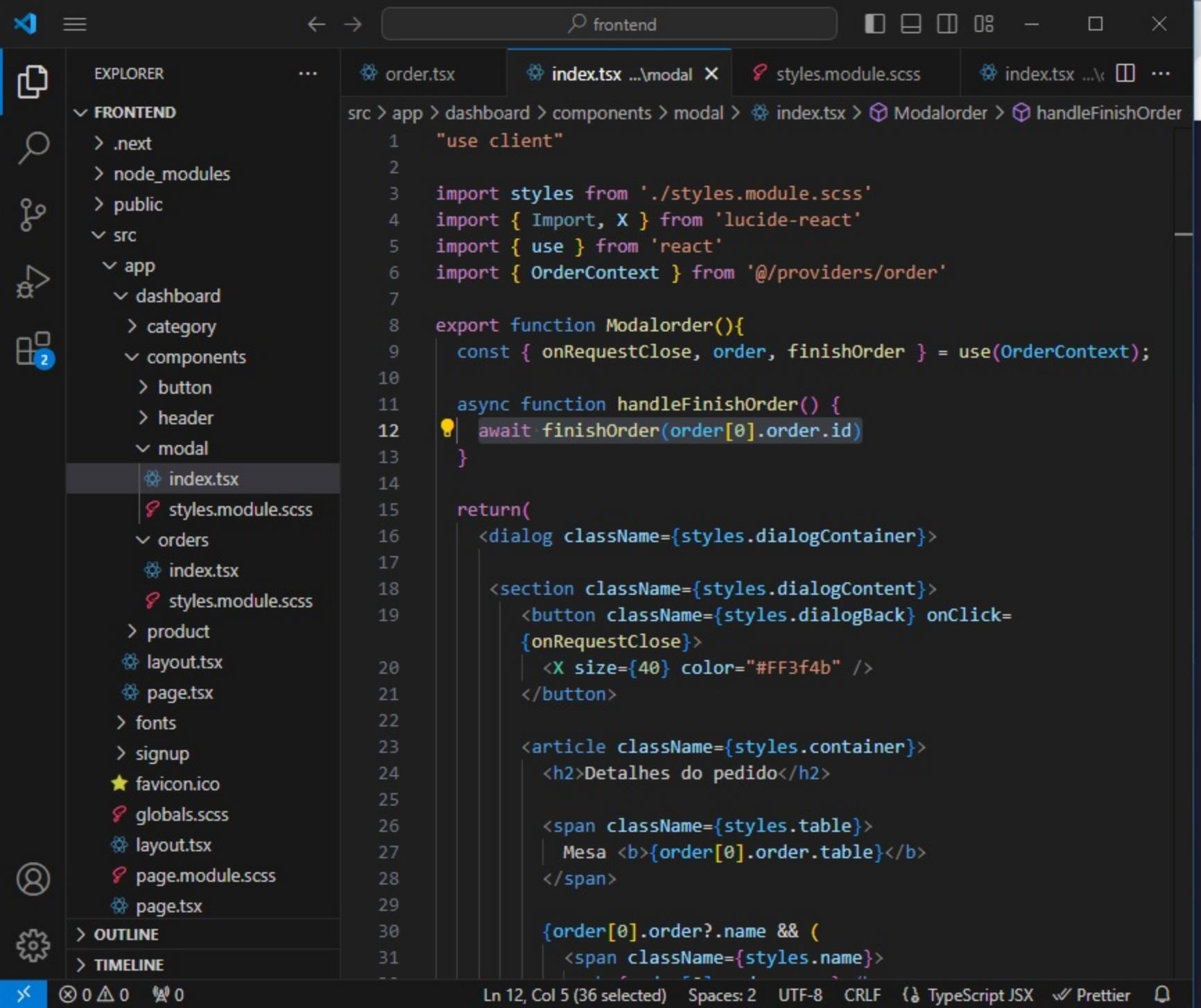
The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- Search:** Shows search results.
- File:** Shows file navigation.
- Terminal:** Shows command-line interface.
- Outline:** Shows code structure.
- Timeline:** Shows code history.
- Editor:** Displays the file `index.tsx` (highlighted) and other files like `order.tsx`, `styles.module.scss`, etc. The code uses TypeScript and SCSS.
- Status Bar:** Shows line 12, column 5, 14 selected, and file type TypeScript/JSX.

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- Address:** localhost:300...
- Content:** Shows the "Últimos pedidos" (Last orders) section with two items:
 - Mesa 35
 - Mesa 55
- Header:** Categories and products.



frontend

EXPLORER

FRONTEND

- > .next
- > node_modules
- > public
- src
 - app
 - dashboard
 - > category
 - > components
 - > button
 - > header
 - > modal
 - index.tsx
 - styles.module.scss
 - > orders
 - index.tsx
 - styles.module.scss
 - > product
 - layout.tsx
 - page.tsx
 - > fonts
 - > signup
 - favicon.ico
 - globals.scss
 - layout.tsx
 - page.module.scss
 - page.tsx

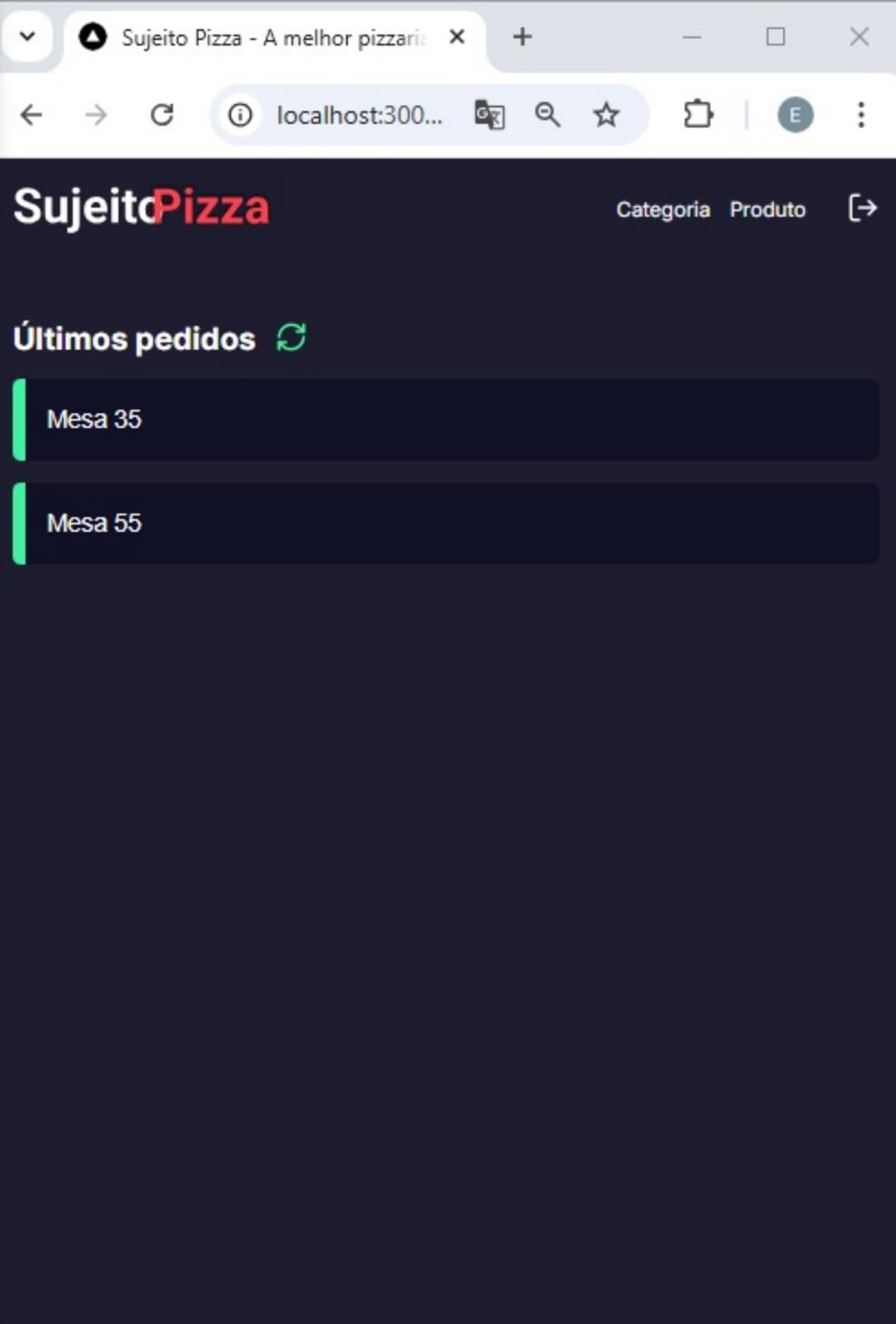
OUTLINE

TIMELINE

order.tsx index.tsx ...\\modal styles.module.scss index.tsx ...\\...
src > app > dashboard > components > modal > index.tsx > Modalorder > handleFinishOrder

```
1 "use client"
2
3 import styles from './styles.module.scss'
4 import { Import, X } from 'lucide-react'
5 import { use } from 'react'
6 import { OrderContext } from '@/providers/order'
7
8 export function Modalorder(){
9   const { onRequestClose, order, finishOrder } = use(OrderContext);
10
11   async function handleFinishOrder() {
12     await finishOrder(order[0].order.id)
13   }
14
15   return(
16     <dialog className={styles.dialogContainer}>
17       <section className={styles.dialogContent}>
18         <button className={styles.dialogBack} onClick={onRequestClose}>
19           <X size={40} color="#FF3f4b" />
20         </button>
21
22         <article className={styles.container}>
23           <h2>Detalhes do pedido</h2>
24
25           <span className={styles.table}>
26             Mesa <b>{order[0].order.table}</b>
27           </span>
28
29           {order[0].order?.name && (
30             <span className={styles.name}>
31               ...
```

Ln 12, Col 5 (36 selected) Spaces: 2 UTF-8 CRLF ⓘ TypeScript JSX ✓ Prettier



Sujeito Pizza - A melhor pizzaria

localhost:300...

SujeitoPizza

Categoria Produto

Últimos pedidos

Mesa 35

Mesa 55

Sujeito Pizza - A melhor pizzaria

localhost:3000/dashboard

Categoria Produto

SujeitoPizza

Últimos pedidos

- Mesa 35
- Mesa 55

```
order.tsx index.tsx
src > app > dashboard > component
1 "use client"
2
3 import styles from
4 import { Import, X
5 import { use } from
6 import { OrderConte
7
8 export function Mod
9 const { onRequestC
10
11 async function ha
12 | await finishOrd
13 }
14
15 return(
16 | <dialog classNa
17 | <section class
18 | <button cla
19 | {onRequestC
20 | | <X size={
21 | </button>
22
23 <article cl
24 <h2>Detal
25
26 <span cla
27 | Mesa <b>
28 </span>
29
30 {order[0]
31 <span c
```

Ln 12, Col 5 (36 selected)

FRONTEND

- .next
- node_modules
- public
- src
 - app
 - category
 - components
 - button
 - header
 - modal
 - index.tsx
 - styles.module.scss
 - orders
 - index.tsx
 - styles.module.scss
 - product
 - layout.tsx
 - page.tsx
 - fonts
 - signup
 - favicon.ico
 - globals.scss
 - layout.tsx
 - page.module.scss
 - page.tsx

OUTLINE

TIMELINE

File Edit View Ultimate Tools Help localhost:5432/pizzaria - Beekeeper Studio

pizzaria C +

Query #1 products [all] orders [all] x +

Filter Y

ENTITIES 217 +

public

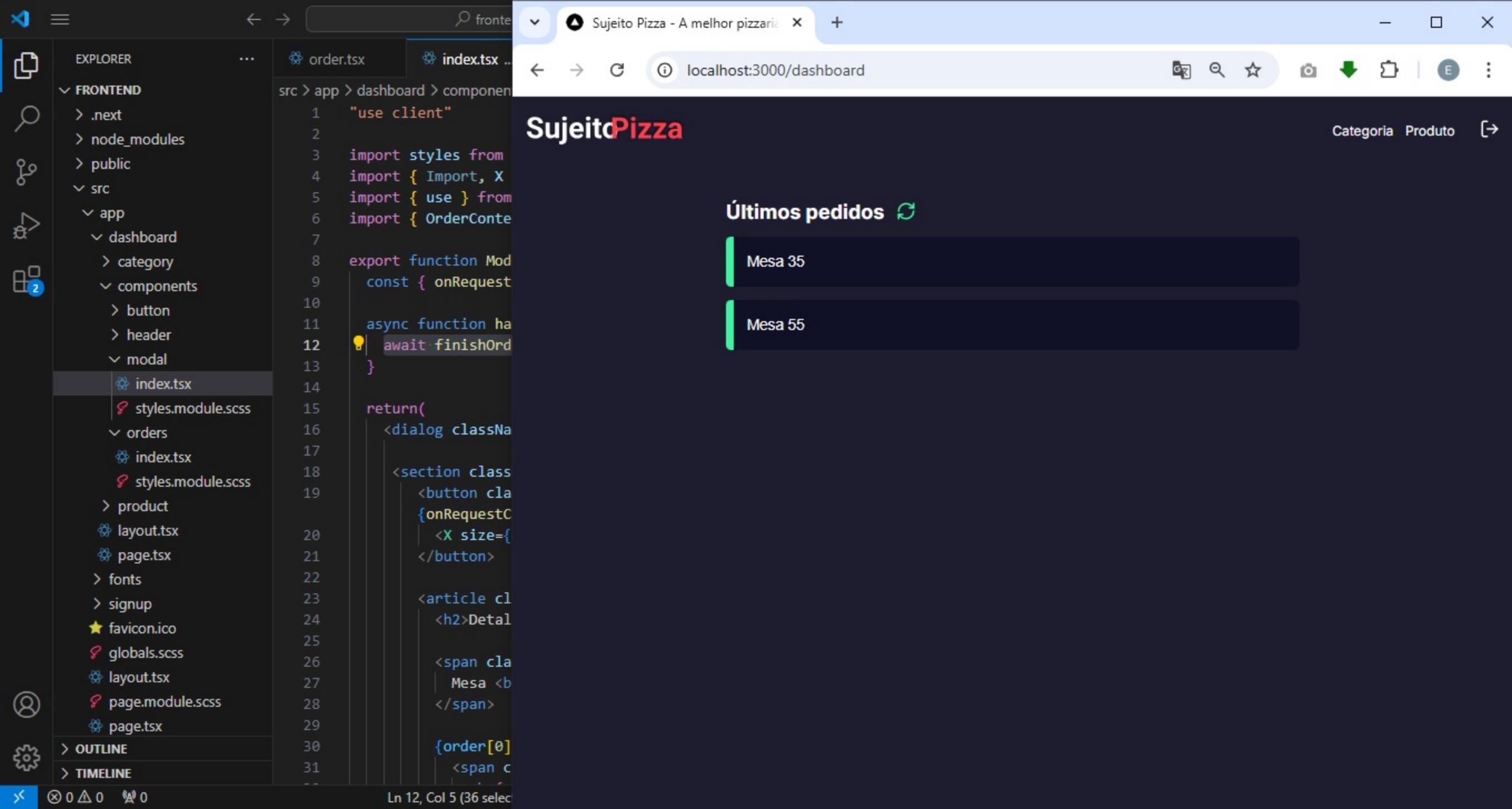
- _prisma_migrations
- categories
- items
- orders
- products
- users

information_schema

pg_catalog

id table status draft name created_at update_at

| | id | table | status | draft | name | created_at | update_at |
|---|---------------------------------------|-------|--------|-------|---------|----------------|--------------------------|
| 1 | 0d0fef8f-9947-4e0a-94d2-dc2722b9cee0 | 55 | false | false | (NULL) | 2024-11-26 ... | 2024-11-26 01:30:26.0... |
| 2 | c9019c83-9a02-4152-a475-21f07e1e39... | 9 | true | false | (NULL) | 2024-11-26 ... | 2024-11-26 15:36:31.8... |
| 3 | ee42b24f-cdc5-48fd-87d3-1758a06afae3 | 35 | false | false | (EMPTY) | 2024-11-29 ... | 2024-11-29 02:28:31.9... |



The screenshot shows a developer's environment with a code editor and a browser preview.

Code Editor:

- EXPLORER:** Shows the project structure. The **FRONTEND** folder is expanded, showing **.next**, **node_modules**, **public**, and **src**. The **src** folder contains **app**, which further contains **dashboard**, **category**, **components**, **button**, **header**, and **modal**. The **index.tsx** file under **dashboard** is currently selected.
- order.tsx** and **index.tsx** are also visible in the editor.
- OUTLINE** and **TIMELINE** are listed at the bottom of the Explorer.

Browser Preview:

- Title Bar:** "Sujeito Pizza - A melhor pizzaria" and "localhost:3000/dashboard".
- Header:** "Sujeito Pizza" and "Dashboard".
- Content:** A modal dialog titled "Detalhes do pedido" (Order Details) is displayed. It shows:
 - Mesa 55**
 - 2 - Coca lata**
Coca-Cola lata 350ml
 - 3 - Pizza de calabresa**
Pizza de calabresa saborosa
- Buttons:** A green "Concluir pedido" (Finish order) button.

Bottom Status Bar: "Ln 12, Col 5 (36 selected)"

Sujeito Pizza - A melhor pizzaria

localhost:3000/dashboard

SupertoPizza

Detalhes do pedido

Mesa 55

2 - Coca lata
Coca-Cola lata 350ml

3 - Pizza de calabresa
Pizza de calabresa saborosa

Concluir pedido

Pedido finalizado com sucesso!

```
order.tsx index.tsx
src > app > dashboard > component
  1 "use client"
  2
  3 import styles from
  4 import { Import, X
  5 import { use } from
  6 import { OrderConte
  7
  8 export function Mod
  9   const { onRequest
 10
 11   async function ha
 12     await finishOrd
 13
 14
 15   return(
 16     <dialog classNa
 17       <section class
 18         <button cla
 19           {onRequestC
 20             <X size={
 21           </button>
 22
 23           <article cl
 24             <h2>Detal
 25
 26             <span cla
 27               Mesa <b
 28             </span>
 29
 30             {order[0]
 31               <span c
```

Ln 12, Col 5 (36 selected)

File Edit View Ultimate Tools Help localhost:5432/pizzaria - Beekeeper Studio

pizzaria C + Query #1 products [all] orders [all] +

Filter ENTITIES 217 +

public

- _prisma_migrations
- categories
- items
- orders
- products
- users

information_schema

pg_catalog

| | id | table | status | draft | name | created_at... | update_at |
|---|---------------------------------------|-------|--------|-------|---------|----------------|--------------------------|
| 1 | 0d0fef8f-9947-4e0a-94d2-dc2722b9cee0 | 55 | false | false | (NULL) | 2024-11-26 ... | 2024-11-26 01:30:26.0... |
| 2 | c9019c83-9a02-4152-a475-21f07e1e39... | 9 | true | false | (NULL) | 2024-11-26 ... | 2024-11-26 15:36:31.8... |
| 3 | ee42b24f-cdc5-48fd-87d3-1758a06afae3 | 35 | false | false | (EMPTY) | 2024-11-29 ... | 2024-11-29 02:28:31.9... |

File Edit View Ultimate Tools Help localhost:5432/pizzaria - Beekeeper Studio

pizzaria C +

Query #1 products [all] orders [all] x +

Filter Y

ENTITIES 217 +

public

- _prisma_migrations
- categories
- items
- orders
- products
- users

information_schema

pg_catalog

id text table int4(32,0) status bool draft bool name text created_at... update_at timestamp(3)

| | id | table | status | draft | name | created_at... | update_at |
|---|---------------------------------------|-------|--------|-------|---------|----------------|--------------------------|
| 1 | 0d0fef8f-9947-4e0a-94d2-dc2722b9cee0 | 55 | false | false | (NULL) | 2024-11-26 ... | 2024-11-26 01:30:26.0... |
| 2 | c9019c83-9a02-4152-a475-21f07e1e39... | 9 | true | false | (NULL) | 2024-11-26 ... | 2024-11-26 15:36:31.8... |
| 3 | ee42b24f-cdc5-48fd-87d3-1758a06afae3 | 35 | false | false | (EMPTY) | 2024-11-29 ... | 2024-11-29 02:28:31.9... |

Refresh Table (ctrl+r or F5)

File Edit View Ultimate Tools Help localhost:5432/pizzaria - Beekeeper Studio

pizzaria C + Query #1 products [all] orders [all] +

Filter ENTITIES 217 +

public

- _prisma_migrations
- categories
- items
- orders
- products
- users

information_schema

pg_catalog

| | id | table | status | draft | name | created_at... | update_at |
|---|---------------------------------------|-------|--------|-------|---------|----------------|--------------------------|
| 1 | 0d0fef8f-9947-4e0a-94d2-dc2722b9cee0 | 55 | true | false | (NULL) | 2024-11-26 ... | 2024-11-26 01:30:26.0... |
| 2 | c9019c83-9a02-4152-a475-21f07e1e39... | 9 | true | false | (NULL) | 2024-11-26 ... | 2024-11-26 15:36:31.8... |
| 3 | ee42b24f-cdc5-48fd-87d3-1758a06afae3 | 35 | false | false | (EMPTY) | 2024-11-29 ... | 2024-11-29 02:28:31.9... |

Sujeito Pizza - A melhor pizzaria

localhost:3000/dashboard

fronte

EXPLORER

FRONTEND

- .next
- node_modules
- public
- src
 - app
 - category
 - components
 - button
 - header
 - modal
 - index.tsx
 - styles.module.scss
 - orders
 - index.tsx
 - styles.module.scss
 - product
 - layout.tsx
 - page.tsx
 - fonts
 - signup
 - favicon.ico
 - globals.scss
 - layout.tsx
 - page.module.scss
 - page.tsx
- OUTLINE
- TIMELINE

order.tsx

index.tsx

```
1 "use client"
2
3 import styles from
4 import { Import, X
5 import { use } from
6 import { OrderConte
7
8 export function Mod
9 const { onRequest
10
11 async function ha
12 | await finishOrd
13 }
14
15 return(
16 | <dialog classNa
17 | <section class
18 | | <button cla
19 | | {onRequestC
20 | | | <X size={
21 | | | </button>
22
23 | <article cl
24 | | <h2>Detal
25
26 | | <span cla
27 | | | Mesa <b
28 | | </span>
29
30 | {order[0]
31 | <span c
```

Ln 12, Col 5 (36 selected)

SupertoPizza

Detalhes do pedido

Mesa 55

2 - Coca lata

Coca-Cola lata 350ml

3 - Pizza de calabresa

Pizza de calabresa saborosa

Concluir pedido

Sujeito Pizza - A melhor pizzaria

localhost:3000/dashboard

Categoria Produto []

SujeitoPizza

Últimos pedidos

Mesa 35

Mesa 55

```
order.tsx index.tsx
src > app > dashboard > components
  1 "use client"
  2
  3 import styles from
  4 import { Import, X
  5 import { use } from
  6 import { OrderConte
  7
  8 export function Mod
  9   const { onRequestC
 10
 11   async function ha
 12     await finishOrd
 13
 14
 15   return(
 16     <dialog className="d
 17       <section className="d
 18         <button className="d
 19           &onRequestC
 20             <X size={&
 21               </button>
 22
 23             <article className="d
 24               <h2>Detal
 25
 26               <span className="d
 27                 Mesa <b>
 28               </span>
 29
 30               {order[0]
 31                 <span>
```

Ln 12, Col 5 (36 selected)

A screenshot of a developer's workspace, likely using a combination of VS Code and a browser. The left side shows the VS Code interface with the Explorer, Search, and Outline/Timeline panes. The Explorer pane displays a file tree for a 'FRONTEND' project, including '.next', 'node_modules', 'public', 'src', 'app', 'dashboard', 'category', 'components', 'button', 'header', 'modal', 'index.tsx', 'styles.module.scss', 'orders', 'index.tsx', 'styles.module.scss', 'product', 'layout.tsx', 'page.tsx', 'fonts', 'signup', 'favicon.ico', 'globals.scss', 'layout.tsx', 'page.module.scss', and 'page.tsx'. The 'index.tsx' file is currently selected in the Explorer.

The main area shows a code editor with the file 'index.tsx' open. The code is a React component that renders a dialog with sections for 'Últimos pedidos' and 'Mesa 35'. A yellow lightbulb icon is visible on line 12, indicating a potential issue or suggestion.

The browser preview at the top shows the application running at 'localhost:3000/dashboard'. The page title is 'Sujeito Pizza - A melhor pizzaria'. The dashboard displays the heading 'SujeitoPizza' and a section titled 'Últimos pedidos' with a refresh icon. Below it is a card for 'Mesa 35'.

At the bottom, the status bar indicates 'Ln 12, Col 5 (36 selected)'.

```
order.tsx index.tsx
src > app > dashboard > components
1 "use client"
2
3 import styles from
4 import { Import, X
5 import { use } from
6 import { OrderConte
7
8 export function Mod
9 const { onRequestC
10
11 async function ha
12 | await finishOrd
13
14
15 return(
16   <dialog className="d
17     <section className="d
18       <button className="d
19         &onRequestClose={&
20           <X size={&
21             </button>
22
23       <article className="d
24         <h2>Detalh
25
26         <span className="d
27           Mesa <b>
28         </span>
29
30         {order[0]
31           <span>
```

frontend

EXPLORER

FRONTEND

- src
 - providers
 - order.tsx
 - index.tsx ...\\modal
 - styles.module.scss
 - index.tsx ...\\

src > providers > order.tsx > OrderProvider > finishOrder

```
function onRequestClose(){
  setIsOpen(false);
}

async function finishOrder(order_id: string) {
  const token = getCookieClient();

  const data = {
    order_id: order_id,
  }

  try {
    await api.put("/order/finish", data, {
      headers: {
        Authorization: `Bearer ${token}`
      }
    })
  } catch(err){
    console.log(err);
    toast.error("Falha ao finalizar este pedido!")
    return;
  }

  toast.success("Pedido finalizado com sucesso!")
  setIsOpen(false);
}

return(
  <OrderContext.Provider
    value={{
      isOpen,
```

Ln 90, Col 5 (17 selected) Spaces: 2 UTF-8 CRLF TypeScript JSX Prettier

Sujeito Pizza - A melhor pizzaria

localhost:3000/das...

SujeitoPizza

Categoria Produto [→]

Últimos pedidos

Mesa 35

The image shows a developer's workspace with two main windows: a code editor and a browser preview.

Code Editor (Left):

- EXPLORER:** Shows the project structure under **FRONTEND**, including **src**, **app**, **dashboard**, **components**, **orders**, **product**, **fonts**, **signup**, **favicon.ico**, **globals.scss**, **layout.tsx**, **page.tsx**, **lib**, **providers**, **order.tsx** (selected), **services**, **middleware.ts**, **.gitignore**, **next-env.d.ts**, **next.config.ts**, and **package-lock.json**.
- order.tsx:** The selected file contains TypeScript code defining an interface **OrderItemProps** and a type **OrderContextData**.

```
order.tsx
1 "use client"
2 import { createContext, ReactNode, useState } from 'react'
3 import { api } from '@/services/api';
4 import { getCookieClient } from '@/lib/cookieClient';
5 import { toast } from 'sonner';
6 import { useRouter } from 'next/navigation';
7
8 interface OrderItemProps{
9   id: string;
10  amount: number;
11  created_at: string;
12  order_id: string;
13  product_id: string;
14  product:{
15    id: string;
16    name: string;
17    price: string;
18    description: string;
19    banner: string;
20    category_id: string;
21  };
22  order:{
23    id: string;
24    table: number;
25    name: string | null;
26    draft: boolean;
27    status: boolean;
28  }
29 }
30
31 type OrderContextData = {
32   isOpen: boolean;
}
```

Browser Preview (Right):

The browser window displays the application's homepage for "Sujeito Pizza". The page features a header with the logo and navigation links for **Categoria** and **Produto**. Below the header, there is a section titled **Últimos pedidos** with a link to **Mesa 35**.

```
order.tsx
src > providers > order.tsx > OrderProvider
27     status: boolean;
28   }
29 }
30
31 type OrderContextData = {
32   isOpen: boolean;
33   onRequestOpen: (order_id: string) => Promise<void>;
34   onRequestClose: () => void;
35   order: OrderItemProps[];
36   finishOrder: (order_id: string) => Promise<void>;
37 }
38
39 type OrderProviderProps = {
40   children: ReactNode;
41 }
42
43 export const OrderContext = createContext({} as OrderContextData)
44
45 export function OrderProvider({ children }: OrderProviderProps){
46   const [isOpen, setIsOpen] = useState(false);
47   const [order, setOrder] = useState<OrderItemProps[]>([]);
48   const router = useRouter();
49
50   async function onRequestOpen(order_id: string){
51     // console.log(order_id);
52
53     const token = getCookieClient();
54
55     const response = await api.get("/order/detail", {
56       headers: {
57         Authorization: `Bearer ${token}`
58       },
59       params: {
```

SujeitoPizza

Categoria Produto

Últimos pedidos

Mesa 35

```
order.tsx
src > providers > order.tsx > OrderProvider > finishOrder
72  async function finishOrder(order_id: string) {
73    const token = getCookieClient();
74
75    const data = {
76      order_id: order_id,
77    }
78
79    try {
80      await api.put("/order/finish", data, {
81        headers: {
82          Authorization: `Bearer ${token}`
83        }
84      })
85    } catch(err){
86      console.log(err);
87      toast.error("Falha ao finalizar este pedido!")
88      return;
89    }
90
91    toast.success("Pedido finalizado com sucesso!")
92    router.refresh();
93    setIsOpen(false);
94  }
95
96
97  return(
98    <OrderContext.Provider
99      value={{
100        isOpen,
101        onRequestOpen,
102        onRequestClose,
103        finishOrder,
104      }}
105    >
```

SujeitoPizza

Últimos pedidos [⟳](#)

Mesa 35

localhost:3000/das...

A screenshot of a developer's workspace, likely using a tool like VS Code or a similar integrated development environment (IDE). The interface is split into several panes:

- Left Sidebar:** Shows the project structure under "FRONTEND". Key components include "src", "app", "dashboard", "components", "orders", "index.tsx", "styles.module.scss", "product", "layout.tsx", "page.tsx", "fonts", "signup", "favicon.ico", "globals.scss", "layout.tsx", "page.module.scss", "page.tsx", "lib", "providers", "order.tsx" (which is currently selected), "services", "middleware.ts", ".gitignore", "next-env.d.ts", "next.config.ts", "package-lock.json", and "OUTLINE/TIMELINE".
- Top Center:** A search bar labeled "frontend".
- Top Right:** A browser window titled "Sujeito Pizza - A melhor pizzaria" showing the URL "localhost:3000/dashboard". The page content includes the text "SujeitoPizza" and "Últimos pedidos" (Last orders) followed by a card for "Mesa 35".
- Bottom:** Status bar showing "Ln 92, Col 5 (17 selected)" and "Spaces: 2 UTF-8".

```
async function finishOrder(order_id) {
  const token = getCookieClient();
  const data = {
    order_id: order_id,
  };
  try {
    await api.put("/order/finish",
      headers: {
        Authorization: `Bearer ${token}`
      }
    );
  } catch(err){
    console.log(err);
    toast.error("Falha ao finalizar pedido");
    return;
  }
  toast.success("Pedido finalizado com sucesso!");
  router.refresh();
  setIsOpen(false);
}

return(
  <OrderContext.Provider
    value={{
      isOpen,
      onRequestOpen,
      onRequestClose,
      finishOrder,
    }}>
```

The screenshot shows a developer's environment with a code editor and a browser preview.

Code Editor:

- EXPLORER:** Shows the project structure under **FRONTEND**, including **src**, **app**, **dashboard**, **components**, **orders**, **index.tsx**, **styles.module.scss**, **product**, **layout.tsx**, **page.tsx**, **fonts**, **signup**, **favicon.ico**, **globals.scss**, **layout.tsx**, **page.module.scss**, **page.tsx**, **lib**, **providers**, **order.tsx**, **services**, **middleware.ts**, **.gitignore**, **next-env.d.ts**, **next.config.ts**, and **package-lock.json**.
- order.tsx:** The active file contains the following code:

```
72  async function finishOrder(order_id) {
73    const token = getCookieClient();
74
75    const data = {
76      order_id: order_id,
77    }
78
79    try {
80      await api.put("/order/finish",
81        headers: {
82          Authorization: `Bearer ${token}`
83        }
84      )
85    } catch(err){
86      console.log(err);
87      toast.error("Falha ao finalizar pedido");
88      return;
89    }
90
91    toast.success("Pedido finalizado com sucesso!");
92    router.refresh();
93    setIsOpen(false);
94
95  }
96
97  return(
98    <OrderContext.Provider
99      value={{
100        isOpen,
101        onRequestOpen,
102        onRequestClose,
103        finishOrder,
104      }}
105    >
106      {children}
107    </OrderContext.Provider>
108  );
109}
```

Browser Preview:

The browser window displays the application at localhost:3000/dashboard. A modal dialog titled "Detalhes do pedido" is open, showing the following information:

- Mesa 35**
- 2 - Burger Triplo Dev**
Hamburger Triplo com Cheddar
- 1 - Dev Burger Salada**
Burgão Dev de Salada com 1 hambúrguer de 160g

A green button labeled "Concluir pedido" is visible at the bottom of the modal.

A screenshot of a developer's workspace, likely using a tool like VS Code or a similar integrated development environment (IDE). The interface is split into several panes:

- Left Sidebar:** Shows the project structure under "EXPLORER". The "FRONTEND" folder is expanded, showing "src", "app", "dashboard", "components", "orders", "product", "layout.tsx", "page.tsx", "fonts", "signup", "favicon.ico", "globals.scss", "layout.tsx", "page.module.scss", "page.tsx", "lib", "providers", and "order.tsx".
- Code Editor:** The "order.tsx" file is open in the editor. The code handles finishing an order by sending a PUT request to the API. It includes error handling with toast notifications for success and failure.
- Browser Preview:** A browser window titled "Sujeito Pizza - A melhor pizzaria" is open at "localhost:3000/dashboard". The page displays the "SujeitoPizza" logo and a section titled "Últimos pedidos" (Last orders) with a refresh icon.
- Status Bar:** At the bottom, the status bar shows "Ln 92, Col 5 (17 selected) Spaces: 2 UTF-8".
- Bottom Right:** A notification bar indicates "Pedido finalizado com sucesso!" (Order completed successfully).

A screenshot of a developer's workspace, likely using a tool like VS Code or a similar integrated development environment (IDE). The interface is split into several panes:

- Left Sidebar:** Shows the project structure under "FRONTEND". The "src" folder contains "app", "dashboard", "components", "orders", "product", "layout.tsx", "page.tsx", "fonts", "signup", "favicon.ico", "globals.scss", "layout.tsx", "page.module.scss", "page.tsx", "lib", "providers", and "order.tsx".
- Center Editor:** Displays the content of the "order.tsx" file. The code is written in TypeScript and handles finishing an order by sending a PUT request to the API. It includes error handling with toast notifications and router refresh.
- Right Preview:** Shows a screenshot of the application running at "localhost:3000/dashboard". The app has a header with the logo "SujeitoPizza". Below it is a section titled "Últimos pedidos" (Last orders) with a circular refresh icon.
- Bottom Status Bar:** Shows file statistics: "Ln 92, Col 5 (17 selected)", "Spaces: 2", and "UTF-8".

UC: Projetos

Assunto:

Projeto Pizzaria – Frontend_02 (NextJS)

Sistema Pizzaria Web

63 – Ajustando Dashboard

The image shows a developer's workspace with two main windows: a code editor and a browser preview.

Code Editor (Left):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "page.tsx" is the active file.
- Content:** The code is an asynchronous function named "Dashboard" that fetches orders from an API and returns them.

```
const response = await api.get("/orders", {
  headers: {
    Authorization: `Bearer ${token}`
  }
});
return response.data || [];

} catch(err){
  console.log(err);
  return [];
}

export default async function Dashboard(){
  const orders = await getOrders();

  return(
    <>
      <Orders orders={orders}>/>
    </>
  )
}
```

Browser Preview (Right):

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localhost:3000/das...
- Header:** SujeitoPizza
- Section:** Últimos pedidos

frontend

index.tsx

```
src > app > dashboard > components > orders > index.tsx > Orders
  19
  20
  21     return(
  22       <>
  23         <main className={styles.container}>
  24           <section className={styles.containerHeader}>
  25             <h1>Últimos pedidos</h1>
  26             <button>
  27               <RefreshCw size={24} color="#3ffffa3" />
  28             </button>
  29           </section>
  30
  31           <section className={styles.listOrders}>
  32             {orders.map( order => (
  33               <button
  34                 key={order.id}
  35                 className={styles.orderItem}
  36                 onClick={() => handleDetailOrder(order.id)}
  37               >
  38                 <div className={styles.tag}></div>
  39                 <span>Mesa {order.table}</span>
  40               </button>
  41             )));
  42           </section>
  43         </main>
  44
  45         { isOpen && <Modalorder/> }
  46       </>
  47     )
  48   )
  49 }
```

Ln 33, Col 1 (339 selected) Spaces: 2 CRLF TypeScript JSX Prettier

Sujeito Pizza - A melhor pizzaria

localhost:3000/das...

SujeitoPizza

Categoria Produto

Últimos pedidos

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file, showing TypeScript code for rendering a dashboard component.
- Content:** The code includes a header section with a refresh button and a list section. The list section contains logic to check if there are no orders. If true, it displays a message: "Nenhum pedido aberto no momento..." (No open orders at the moment...).
- Bottom Status Bar:** Shows file path, line/col count (Ln 33, Col 1 (169 selected)), and file type (TypeScript JSX).

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localhost:3000/dashboard
- Content:** Displays the "SujeitoPizza" logo and a heading "Últimos pedidos". Below it, a message states "Nenhum pedido aberto no momento...".
- Header:** Includes links for "Categoria", "Produto", and a search icon.

The image shows a developer's workspace with two main windows: a code editor and a web browser.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file.
- File:** "styles.module.scss" is also open.
- Content:** The "styles.module.scss" file contains SCSS code for a component named "orders". It includes styles for a button-like element with a hover effect, a ".tag" class with specific dimensions and colors, and a ".emptyItem" class with a gray color.

Browser:

- Title:** "Sujeito Pizza - A melhor pizzaria"
- URL:** "localhost:3000/das..."
- Content:** The page displays the "SujeitoPizza" logo and a section titled "Últimos pedidos" which states "Nenhum pedido aberto no momento...".

At the bottom of the screen, there are status bars for both the code editor and the browser, showing file paths, line numbers, and other relevant information.

```
src > app > dashboard > components > orders > styles.module.scss > ...
38     border:0;
39     font-size: 18px;
40     border-radius: 8px;
41     color: var(--white);
42     transition: all 0.5s;
43
44     &:hover{
45       filter: brightness(1.40);
46     }
47   }
48
49   .tag{
50     width: 9px;
51     height: 60px;
52     background-color: var(--green-900);
53     border-radius: 6px 0 0 6px;
54     margin-right: 16px;
55   }
56
57   .emptyItem {
58     color: var(--gray-100);
59   }
```

Ln 57, Col 1 (48 selected) Spaces: 2 UTF-8 CRLF { SCSS ✓ Prettier

The image shows a developer's workspace with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file.
- Content:** The code is a TypeScript component for displaying orders. It includes imports from "src/styles.module.scss", a section for "Últimos pedidos", and a map function that loops through "orders" to render individual order details.

```
src > app > dashboard > components > orders > index.tsx > Orders
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
```

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localhost:3000/das...
- Content:** Displays the heading "Últimos pedidos" and a message indicating "Nenhum pedido aberto no momento...".

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file.
- Content:** The code is a component named "Orders" that displays the last orders. It uses TypeScript, React, and Context API. It includes imports for OrderProps, Modalorder, OrderContext, useRouter, and toast. It defines a Props interface with an orders array and an export function Orders. The component contains logic for handling order details and refreshing the list.
- Status Bar:** Shows 4 selections (236 characters selected), Spaces: 2, UTF-8, CRLF, TypeScript JSX, Prettier, and a refresh icon.

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localhost:3000/das...
- Content:** Displays the "SujeitoPizza" logo and a section titled "Últimos pedidos" which says "Nenhum pedido aberto no momento...".
- Header:** Includes links for "Categoria" and "Produto" and a search bar.

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file.
- Content:** The code is a TypeScript component named "Orders". It imports "OrderProps" from "@lib/order.type", "Modalorder" from "../modal", "OrderContext" from "@providers/order", "useRouter" from "next/navigation", and "toast" from "sonner".

```
src > app > dashboard > components > orders > index.tsx > Orders
6  import { OrderProps } from '@/lib/order.type'
7  import { Modalorder } from '../modal'
8  import { OrderContext } from '@providers/order'
9  import { useRouter } from 'next/navigation'
10 import { toast } from 'sonner'

11 interface Props{
12   orders: OrderProps[]
13 }

14 export function Orders({ orders }: Props){
15   const { isOpen, onRequestOpen } = use(OrderContext)
16   const router = useRouter()

17   async function handleDetailOrder(order_id: string) {
18     await onRequestOpen(order_id)
19   }

20   function handleRefresh(){
21     router.refresh()
22     toast.success("Pedidos atualizados com sucesso!")
23   }

24   return(
25     <>
26       <main className={styles.container}>
27         <section className={styles.containerHeader}>
28           <h1>Últimos pedidos</h1>
29           <button onClick={handleRefresh}>
30             <RefreshCw size={24} color="#3ffffa3" />
31           </button>
32         </section>
33       </main>
34     </>
35   )
36 }

37 </>
```

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localhost:3000/das...
- Content:** The page displays the title "SujeitoPizza" and a section titled "Últimos pedidos" which says "Nenhum pedido aberto no momento...". A success toast message at the bottom right says "Pedidos atualizados com sucesso!".

A screenshot of a developer's environment, likely Microsoft Visual Studio Code, showing a code editor and a browser window side-by-side.

The code editor on the left displays the file `index.tsx` from a Next.js application. The file contains TypeScript code for a dashboard component, including imports for `OrderProcessor`, `ModalOrder`, `OrderConfirmation`, `useRouter`, and `toast`. It defines a props interface and an export function for handling orders. The browser window on the right shows the application running at `localhost:3000/dashboard`. The page title is "Sujeito Pizza - A melhor pizzaria". The main content area features the "SujeitoPizza" logo and a section titled "Últimos pedidos" with a message "Nenhum pedido aberto no momento...". A success toast notification at the bottom right says "Pedidos atualizados com sucesso!".

```
index.tsx
src > app > dashboard > components > index.tsx
6   import { OrderProcessor } from 'orderprocessor';
7   import { ModalOrder } from 'modalorder';
8   import { OrderConfirmation } from 'orderconfirmation';
9   import { useRouter } from 'next/router';
10  import { toast } from 'react-toastify';
11
12  interface Props {
13    orders: OrderProcessor[];
14  }
15
16  export function Index({ orders }: Props) {
17    const [isOpen, setIsOpen] = useState(false);
18    const router = useRouter();
19
20    async function handleOrder(order) {
21      await onRequestOrder(order);
22      setIsOpen(true);
23    }
24
25    function handleConfirmation() {
26      router.refresh();
27      toast.success('Pedidos atualizados com sucesso!');
28    }
29
30    return (
31      <>
32        <main className="flex flex-col h-full">
33          <section className="flex-grow p-4">
34            <h1>Últimos pedidos</h1>
35            <button onClick={handleConfirmation}><RefreshIcon/> Atualizar</button>
36            </section>
37          </main>
38        </>
39      </>
40    );
41  }
42
43  export default Index;
```

File Edit View Ultimate Tools Help localhost:5432/pizzaria - Beekeeper Studio

pizzaria C + Query #1 products [all] orders [all] +

Filter ENTITIES 217 +

public

- _prisma_migrations
- categories
- items
- orders
- products
- users

information_schema

pg_catalog

| | id | table | status | draft | name | created_at... | update_at |
|---|---------------------------------------|-------|--------|-------|---------|----------------|--------------------------|
| 1 | 0d0fef8f-9947-4e0a-94d2-dc2722b9cee0 | 55 | true | false | (NULL) | 2024-11-26 ... | 2024-11-26 01:30:26.0... |
| 2 | c9019c83-9a02-4152-a475-21f07e1e39... | 9 | true | false | (NULL) | 2024-11-26 ... | 2024-11-26 15:36:31.8... |
| 3 | ee42b24f-cdc5-48fd-87d3-1758a06afae3 | 35 | false | false | (EMPTY) | 2024-11-29 ... | 2024-11-29 02:28:31.9... |

File Edit View Ultimate Tools Help localhost:5432/pizzaria - Beekeeper Studio

pizzaria C + Query #1 products [all] orders [all] +

Filter

ENTITIES 217 +

- public
 - _prisma_migrations
 - categories
 - items
 - orders
 - products
 - users
- information_schema
- pg_catalog

| | id | table | status | draft | name | created_at... | update_at |
|---|---------------------------------------|-------|--------|-------|---------|----------------|--------------------------|
| 1 | 0d0fef8f-9947-4e0a-94d2-dc2722b9cee0 | 55 | true | false | (NULL) | 2024-11-26 ... | 2024-11-26 01:30:26.0... |
| 2 | c9019c83-9a02-4152-a475-21f07e1e39... | 9 | true | false | (NULL) | 2024-11-26 ... | 2024-11-26 15:36:31.8... |
| 3 | ee42b24f-cdc5-48fd-87d3-1758a06afae3 | 35 | true | false | (EMPTY) | 2024-11-29 ... | 2024-11-29 02:28:31.9... |

Refresh Table (ctrl+r or F5)

File Edit View Ultimate Tools Help localhost:5432/pizzaria - Beekeeper Studio

pizzaria C + Query #1 products [all] orders [all] +

Filter

ENTITIES 217 +

public

- _prisma_migrations
- categories
- items
- orders
- products
- users

information_schema

pg_catalog

id table int4(32,0) status bool draft bool name text created_at... update_at timestamp(3)

| | id | table | status | draft | name | created_at... | update_at |
|---|---------------------------------------|-------|--------|-------|---------|----------------|--------------------------|
| 1 | 0d0fef8f-9947-4e0a-94d2-dc2722b9cee0 | 55 | true | false | (NULL) | 2024-11-26 ... | 2024-11-26 01:30:26.0... |
| 2 | c9019c83-9a02-4152-a475-21f07e1e39... | 9 | true | false | (NULL) | 2024-11-26 ... | 2024-11-26 15:36:31.8... |
| 3 | ee42b24f-cdc5-48fd-87d3-1758a06afae3 | 35 | true | false | (EMPTY) | 2024-11-29 ... | 2024-11-29 02:28:31.9... |

File Edit View Ultimate Tools Help localhost:5432/pizzaria - Beekeeper Studio

pizzaria C + Query #1 products [all] orders [all]

Filter ENTITIES 217 +

public

- _prisma_migrations
- categories
- items
- orders
- products
- users

information_schema

pg_catalog

id table status draft name created_at update_at

| | id | table | status | draft | name | created_at | update_at |
|---|---------------------------------------|-------|--------|-------|---------|----------------|--------------------------|
| 1 | 0d0fef8f-9947-4e0a-94d2-dc2722b9cee0 | 55 | true | false | (NULL) | 2024-11-26 ... | 2024-11-26 01:30:26.0... |
| 2 | c9019c83-9a02-4152-a475-21f07e1e39... | 9 | true | false | (NULL) | 2024-11-26 ... | 2024-11-26 15:36:31.8... |
| 3 | ee42b24f-cdc5-48fd-87d3-1758a06afae3 | 35 | false | false | (EMPTY) | 2024-11-29 ... | 2024-11-29 02:28:31.9... |

File Edit View Ultimate Tools Help localhost:5432/pizzaria - Beekeeper Studio

pizzaria C + Query #1 products [all] orders [all] +

Filter ENTITIES 217 +

public

- _prisma_migrations
- categories
- items
- orders
- products
- users

information_schema

pg_catalog

id table status draft name created_at update_at

| | table | status | draft | name | created_at | update_at |
|---|-------|--------|-------|---------|----------------|--------------------------|
| 1 | 55 | true | false | (NULL) | 2024-11-26 ... | 2024-11-26 01:30:26.0... |
| 2 | 9 | true | false | (NULL) | 2024-11-26 ... | 2024-11-26 15:36:31.8... |
| 3 | 35 | false | false | (EMPTY) | 2024-11-29 ... | 2024-11-29 02:28:31.9... |

File Edit View Ultimate Tools Help localhost:5432/pizzaria - Beekeeper Studio

pizzaria C + Query #1 products [all] orders [all] +

Filter ENTITIES 217 +

public

- _prisma_migrations
- categories
- items
- orders
- products
- users

information_schema

pg_catalog

| | id | table | status | draft | name | created_at... | update_at |
|---|---------------------------------------|-------|--------|-------|---------|----------------|--------------------------|
| 1 | 0d0fef8f-9947-4e0a-94d2-dc2722b9cee0 | 55 | true | false | (NULL) | 2024-11-26 ... | 2024-11-26 01:30:26.0... |
| 2 | c9019c83-9a02-4152-a475-21f07e1e39... | 9 | true | false | (NULL) | 2024-11-26 ... | 2024-11-26 15:36:31.8... |
| 3 | ee42b24f-cdc5-48fd-87d3-1758a06afae3 | 35 | false | false | (EMPTY) | 2024-11-29 ... | 2024-11-29 02:28:31.9... |

File Edit View Ultimate Tools Help localhost:5432/pizzaria - Beekeeper Studio

pizzaria C + Query #1 products [all] orders [all] x +

Filter

ENTITIES 217 +

public

- _prisma_migrations
- categories
- items
- orders
- products
- users

information_schema

pg_catalog

id text table int4(32,0) status bool draft bool name text created_at... update_at timestamp(3)

| | id | table | status | draft | name | created_at... | update_at |
|---|---------------------------------------|-------|--------|-------|---------|----------------|--------------------------|
| 1 | 0d0fef8f-9947-4e0a-94d2-dc2722b9cee0 | 55 | true | false | (NULL) | 2024-11-26 ... | 2024-11-26 01:30:26.0... |
| 2 | c9019c83-9a02-4152-a475-21f07e1e39... | 9 | false | false | (NULL) | 2024-11-26 ... | 2024-11-26 15:36:31.8... |
| 3 | ee42b24f-cdc5-48fd-87d3-1758a06afae3 | 35 | true | false | (EMPTY) | 2024-11-29 ... | 2024-11-29 02:28:31.9... |

File Edit View Ultimate Tools Help localhost:5432/pizzaria - Beekeeper Studio

pizzaria C + Query #1 products [all] orders [all]

Filter ENTITIES 217

public

- _prisma_migrations
- categories
- items
- orders
- products
- users

information_schema

pg_catalog

id equals Enter Value

| | id | table | status | draft | name | created_at... | update_at |
|---|---------------------------------------|-------|--------|-------|---------|----------------|--------------------------|
| 1 | 0d0fef8f-9947-4e0a-94d2-dc2722b9cee0 | 55 | false | false | (NULL) | 2024-11-26 ... | 2024-11-26 01:30:26.0... |
| 2 | c9019c83-9a02-4152-a475-21f07e1e39... | 9 | true | false | (NULL) | 2024-11-26 ... | 2024-11-26 15:36:31.8... |
| 3 | ee42b24f-cdc5-48fd-87d3-1758a06afae3 | 35 | false | false | (EMPTY) | 2024-11-29 ... | 2024-11-29 02:28:31.9... |

+

File Edit View Ultimate Tools Help localhost:5432/pizzaria - Beekeeper Studio

pizzaria C +

Query #1 products [all] orders [all] x +

Filter Y

ENTITIES 217 +

public

- _prisma_migrations
- categories
- items
- orders
- products
- users

information_schema

pg_catalog

| | id | table | status | draft | name | created_at... | update_at |
|---|---------------------------------------|-------|--------|-------|---------|----------------|--------------------------|
| 1 | 0d0fef8f-9947-4e0a-94d2-dc2722b9cee0 | 55 | false | false | (NULL) | 2024-11-26 ... | 2024-11-26 01:30:26.0... |
| 2 | c9019c83-9a02-4152-a475-21f07e1e39... | 9 | true | false | (NULL) | 2024-11-26 ... | 2024-11-26 15:36:31.8... |
| 3 | ee42b24f-cdc5-48fd-87d3-1758a06afae3 | 35 | false | false | (EMPTY) | 2024-11-29 ... | 2024-11-29 02:28:31.9... |

The image shows a developer's workspace with two main windows: a code editor in the foreground and a browser preview in the background.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file.
- Content:** The code is a TypeScript component named "Orders". It imports "OrderProps" from "@lib/order.type", "Modalorder" from "../modal", "OrderContext" from "@providers/order", "useRouter" from "next/navigation", and "toast" from "sonner".

```
src > app > dashboard > components > orders > index.tsx > Orders
6  import { OrderProps } from '@/lib/order.type'
7  import { Modalorder } from '../modal'
8  import { OrderContext } from '@providers/order'
9  import { useRouter } from 'next/navigation'
10 import { toast } from 'sonner'

11 interface Props{
12   orders: OrderProps[]
13 }

14 export function Orders({ orders }: Props){
15   const { isOpen, onRequestOpen } = use(OrderContext)
16   const router = useRouter()

17   async function handleDetailOrder(order_id: string) {
18     await onRequestOpen(order_id)
19   }

20   function handleRefresh(){
21     router.refresh()
22     toast.success("Pedidos atualizados com sucesso!")
23   }

24   return(
25     <>
26       <main className={styles.container}>
27         <section className={styles.containerHeader}>
28           <h1>Últimos pedidos</h1>
29           <button onClick={handleRefresh}>
30             <RefreshCw size={24} color="#3fffa3" />
31           </button>
32         </section>
```

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localhost:3000/das...
- Content:** Displays the "Ultimos pedidos" section of the application, which currently shows "Nenhum pedido aberto no momento...".

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file.
- Content:** The code is a TypeScript component named "Orders". It imports "OrderProps" from "@lib/order.type", "Modalorder" from "../modal", "OrderContext" from "@providers/order", "useRouter" from "next/navigation", and "toast" from "sonner".

```
src > app > dashboard > components > orders > index.tsx > Orders
6  import { OrderProps } from '@/lib/order.type'
7  import { Modalorder } from '../modal'
8  import { OrderContext } from '@providers/order'
9  import { useRouter } from 'next/navigation'
10 import { toast } from 'sonner'

11 interface Props{
12   orders: OrderProps[]
13 }

14 export function Orders({ orders }: Props){
15   const { isOpen, onRequestOpen } = use(OrderContext)
16   const router = useRouter()

17   async function handleDetailOrder(order_id: string) {
18     await onRequestOpen(order_id)
19   }

20   function handleRefresh(){
21     router.refresh()
22     toast.success("Pedidos atualizados com sucesso!")
23   }

24   return(
25     <>
26       <main className={styles.container}>
27         <section className={styles.containerHeader}>
28           <h1>Últimos pedidos</h1>
29           <button onClick={handleRefresh}>
30             <RefreshCw size={24} color="#3fffa3" />
31           </button>
32         </section>
33       </main>
34     </>
35   )
36 }

37 </Orders>
```

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localhost:3000/das...
- Content:** The page displays the heading "SujeitoPizza" and a section titled "Últimos pedidos" with two items: "Mesa 35" and "Mesa 55".
- Toast:** A success toast at the bottom right says "Pedidos atualizados com sucesso!" with a checkmark icon.

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is open in the editor.
- Content:** The code is a TypeScript component named "Orders". It imports "OrderProps" from "@lib/order.type", "Modalorder" from "../modal", "OrderContext" from "@providers/order", "useRouter" from "next/navigation", and "toast" from "sonner".

```
src > app > dashboard > components > orders > index.tsx > Orders
6  import { OrderProps } from '@/lib/order.type'
7  import { Modalorder } from '../modal'
8  import { OrderContext } from '@providers/order'
9  import { useRouter } from 'next/navigation'
10 import { toast } from 'sonner'

11 interface Props{
12   orders: OrderProps[]
13 }

14 export function Orders({ orders }: Props){
15   const { isOpen, onRequestOpen } = use(OrderContext)
16   const router = useRouter()

17   async function handleDetailOrder(order_id: string) {
18     await onRequestOpen(order_id)
19   }

20   function handleRefresh(){
21     router.refresh()
22     toast.success("Pedidos atualizados com sucesso!")
23   }

24   return(
25     <>
26       <main className={styles.container}>
27         <section className={styles.containerHeader}>
28           <h1>Últimos pedidos</h1>
29           <button onClick={handleRefresh}>
30             <RefreshCw size={24} color="#3fffa3" />
31           </button>
32         </section>
```

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localhost:3000/das...
- Content:** A modal window titled "Detalhes do pedido" is displayed. It shows a table of items:

| Mesa | Item | Descrição |
|------|--------------------|-----------------------------|
| 2 | Coca lata | Coca-Cola lata 350ml |
| 3 | Pizza de calabresa | Pizza de calabresa saborosa |

A green button labeled "Concluir pedido" is at the bottom right.

The image shows a developer's environment with two main windows: a code editor in the foreground and a browser preview in the background.

Code Editor (VS Code):

- Explorer:** Shows the project structure under **FRONTEND**. The **orders** folder is expanded, showing **index.tsx** and **styles.module.scss**.
- File:** **index.tsx** is open in the editor.
- Content:** The code is a TypeScript component named **Orders** that displays the last orders and provides a refresh function.

```
src > app > dashboard > components > orders > index.tsx > Orders
6  import { OrderProps } from '@/lib/order.type'
7  import { Modalorder } from '../modal'
8  import { OrderContext } from '@/providers/order'
9  import { useRouter } from 'next/navigation'
10 import { toast } from 'sonner'

11 interface Props{
12   orders: OrderProps[]
13 }

14 export function Orders({ orders }: Props){
15   const { isOpen, onRequestOpen } = use(OrderContext)
16   const router = useRouter();

17   async function handleDetailOrder(order_id: string) {
18     await onRequestOpen(order_id)
19   }

20   function handleRefresh(){
21     router.refresh();
22     toast.success("Pedidos atualizados com sucesso!")
23   }

24   return(
25     <>
26       <main className={styles.container}>
27         <section className={styles.containerHeader}>
28           <h1>Últimos pedidos</h1>
29           <button onClick={handleRefresh}>
30             <RefreshCw size={24} color="#3fffa3" />
31           </button>
32         </section>
33       </main>
34     </>
35   )
36 }

37 </>
```

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localhost:3000/das...
- Content:** A dark-themed web application showing the last orders for table 35. The first order is a "Burger Triplo Dev" (Hamburger Triplo com Cheddar). The second order is a "Dev Burger Salada" (Burgão Dev de Salada com 1 hamburger de 160g). A green button labeled "Concluir pedido" (Finish order) is visible at the bottom right of the modal.

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file.
- Content:** The code is a TypeScript component named "Orders". It imports "OrderProps" from "@lib/order.type", "Modalorder" from "../modal", "OrderContext" from "@providers/order", "useRouter" from "next/navigation", and "toast" from "sonner".

```
src > app > dashboard > components > orders > index.tsx > Orders
6  import { OrderProps } from '@/lib/order.type'
7  import { Modalorder } from '../modal'
8  import { OrderContext } from '@providers/order'
9  import { useRouter } from 'next/navigation'
10 import { toast } from 'sonner'

11 interface Props{
12   orders: OrderProps[]
13 }

14 export function Orders({ orders }: Props){
15   const { isOpen, onRequestOpen } = use(OrderContext)
16   const router = useRouter()

17   async function handleDetailOrder(order_id: string) {
18     await onRequestOpen(order_id)
19   }

20   function handleRefresh(){
21     router.refresh()
22     toast.success("Pedidos atualizados com sucesso!")
23   }

24   return(
25     <>
26       <main className={styles.container}>
27         <section className={styles.containerHeader}>
28           <h1>Últimos pedidos</h1>
29           <button onClick={handleRefresh}>
30             <RefreshCw size={24} color="#3fffa3" />
31           </button>
32         </section>
33       </main>
34     </>
35   )
36 }
```

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localhost:3000/das...
- Content:** The page displays the title "SujeitoPizza" and a section titled "Últimos pedidos" with two items: "Mesa 35" and "Mesa 55".
- Toast:** A success toast at the bottom right says "Pedidos atualizados com sucesso!" with a checkmark icon.

The image shows a developer's environment with two main windows: a code editor and a web browser.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** index.tsx (selected), styles.module.scss, layout.tsx, page.tsx, fonts, signup, favicon.ico, globals.scss, layout.module.scss, page.module.scss, page.tsx, lib, providers.
- Search:** Search bar at the top with the query "frontend".
- Status Bar:** Shows file statistics (Ln 23, Col 1), code style (Spaces: 2, CRLF), and Prettier status.

```
src > app > dashboard > components > orders > index.tsx > Orders
6   import { OrderProps } from '@/lib/order.type'
7   import { Modalorder } from '../modal'
8   import { OrderContext } from '@/providers/order'
9   import { useRouter } from 'next/navigation'
10  import { toast } from 'sonner'

11 interface Props{
12   | orders: OrderProps[]
13 }

14 export function Orders({ orders }: Props){
15   const { isOpen, onRequestOpen } = use(OrderContext)
16   const router = useRouter()

17   async function handleDetailOrder(order_id: string) {
18     await onRequestOpen(order_id)
19   }

20   function handleRefresh(){
21     router.refresh()
22     toast.success("Pedidos atualizados com sucesso!")
23   }

24   return(
25     <>
26       <main className={styles.container}>
27         <section className={styles.containerHeader}>
28           <h1>Últimos pedidos</h1>
29           <button onClick={handleRefresh}>
30             <RefreshCw size={24} color="#3fffa3" />
31           </button>
32         </section>
33       </main>
34   )
35 }
```

Browser:

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localhost:3000/das...
- Content:** A dark-themed web application titled "SujeitoPizza". It has a header with "Categoria" and "Produto" buttons. Below is a section titled "Nova Categoria" with a text input placeholder "Nome da categoria, ex: Pizzas" and a green "Cadastrar" button.

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file.
- Content:** The code is a TypeScript component named "Orders". It imports "OrderProps" from "@lib/order.type", "Modalorder" from "../modal", "OrderContext" from "@providers/order", "useRouter" from "next/navigation", and "toast" from "sonner".

```
src > app > dashboard > components > orders > index.tsx > Orders
6  import { OrderProps } from '@/lib/order.type'
7  import { Modalorder } from '../modal'
8  import { OrderContext } from '@providers/order'
9  import { useRouter } from 'next/navigation'
10 import { toast } from 'sonner'

11 interface Props{
12   orders: OrderProps[]
13 }

14 export function Orders({ orders }: Props){
15   const { isOpen, onRequestOpen } = use(OrderContext)
16   const router = useRouter()

17   async function handleDetailOrder(order_id: string) {
18     await onRequestOpen(order_id)
19   }

20   function handleRefresh(){
21     router.refresh();
22     toast.success("Pedidos atualizados com sucesso!")
23   }

24   return(
25     <>
26       <main className={styles.container}>
27         <section className={styles.containerHeader}>
28           <h1>Últimos pedidos</h1>
29           <button onClick={handleRefresh}>
30             <RefreshCw size={24} color="#3fffa3" />
31           </button>
32         </section>
```

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localhost:3000/das...
- Content:** A dark-themed web page titled "SujeitoPizza" with a sub-section "Novo produto". It features a large input field with a circular arrow icon, and four smaller input fields for "bebidas", "Digite o nome do produto...", "Preço do produto...", and "Digite a descrição do produto...". A green button at the bottom right says "Cadastrar produto".

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file.
- Content:** The code is a TypeScript component named "Orders". It imports "OrderProps" from "@lib/order.type", "Modalorder" from "../modal", "OrderContext" from "@providers/order", "useRouter" from "next/navigation", and "toast" from "sonner".

```
src > app > dashboard > components > orders > index.tsx > Orders
6  import { OrderProps } from '@/lib/order.type'
7  import { Modalorder } from '../modal'
8  import { OrderContext } from '@providers/order'
9  import { useRouter } from 'next/navigation'
10 import { toast } from 'sonner'

11 interface Props{
12   orders: OrderProps[]
13 }

14 export function Orders({ orders }: Props){
15   const { isOpen, onRequestOpen } = use(OrderContext)
16   const router = useRouter()

17   async function handleDetailOrder(order_id: string) {
18     await onRequestOpen(order_id)
19   }

20   function handleRefresh(){
21     router.refresh()
22     toast.success("Pedidos atualizados com sucesso!")
23   }

24   return(
25     <>
26       <main className={styles.container}>
27         <section className={styles.containerHeader}>
28           <h1>Últimos pedidos</h1>
29           <button onClick={handleRefresh}>
30             <RefreshCw size={24} color="#3fffa3" />
31           </button>
32         </section>
33       </main>
34     </>
35   )
36 }
```

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localhost:3000/das...
- Content:** A dark-themed web page titled "SujeitoPizza" with a sub-section "Novo produto". It features a large input field with a refresh icon, and four input fields for "bebidas": "Digite o nome do produto...", "Preço do produto...", "Digite a descrição do produto...", and a green "Cadastrar produto" button.

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is the active file.
- Content:** The code is a TypeScript component named "Orders". It imports "OrderProps" from "@/lib/order.type", "Modalorder" from "../modal", "OrderContext" from "@/providers/order", "useRouter" from "next/navigation", and "toast" from "sonner".

```
src > app > dashboard > components > orders > index.tsx > Orders
6  import { OrderProps } from '@/lib/order.type'
7  import { Modalorder } from '../modal'
8  import { OrderContext } from '@/providers/order'
9  import { useRouter } from 'next/navigation'
10 import { toast } from 'sonner'

11 interface Props{
12   orders: OrderProps[]
13 }

14 export function Orders({ orders }: Props){
15   const { isOpen, onRequestOpen } = use(OrderContext)
16   const router = useRouter()

17   async function handleDetailOrder(order_id: string) {
18     await onRequestOpen(order_id)
19   }

20   function handleRefresh(){
21     router.refresh()
22     toast.success("Pedidos atualizados com sucesso!")
23   }

24   return(
25     <>
26       <main className={styles.container}>
27         <section className={styles.containerHeader}>
28           <h1>Últimos pedidos</h1>
29           <button onClick={handleRefresh}>
30             <RefreshCw size={24} color="#3ffffa3" />
31           </button>
32         </section>
33       </main>
34     </>
35   )
36 }
```

Browser Preview:

- Title:** Sujeito Pizza - A melhor pizzaria
- URL:** localhost:3000
- Content:** Displays the "SujeitoPizza" logo, an email input field ("Digite seu email..."), a password input field ("*****"), and a "Acessar" button. Below the form is a link "Não possui uma conta? Cadastre-se".
- Status Bar:** Shows a success message "Logout Realizado com Sucesso!" with a checkmark icon.

<localhost:3000/dashboard/><localhost:3000/dashboard/>[localhost:3000/dashboard/ - Pesquisa do Google](#)[Sujeito Pizza - A melhor pizzaria - localhost:3000/dashboard/category](#)[Sujeito Pizza - A melhor pizzaria - localhost:3000/dashboard/product](#)[Sujeito Pizza - A melhor pizzaria - localhost:3000/dashboard/product?category=6&name=Açaí+300ml&price=12&description=Pote+de+Açaí+300ml](#)

SujeitoPizza

Digite seu email...*****Acessar

Não possui uma conta? [Cadastre-se](#)

Sujeito**Pizza**

Digite seu email...

Não possui uma conta? [Cadastre-se](#)

SujeitoPizza

Criando sua conta

Digite seu nome...

Digite seu email...

Cadastrar

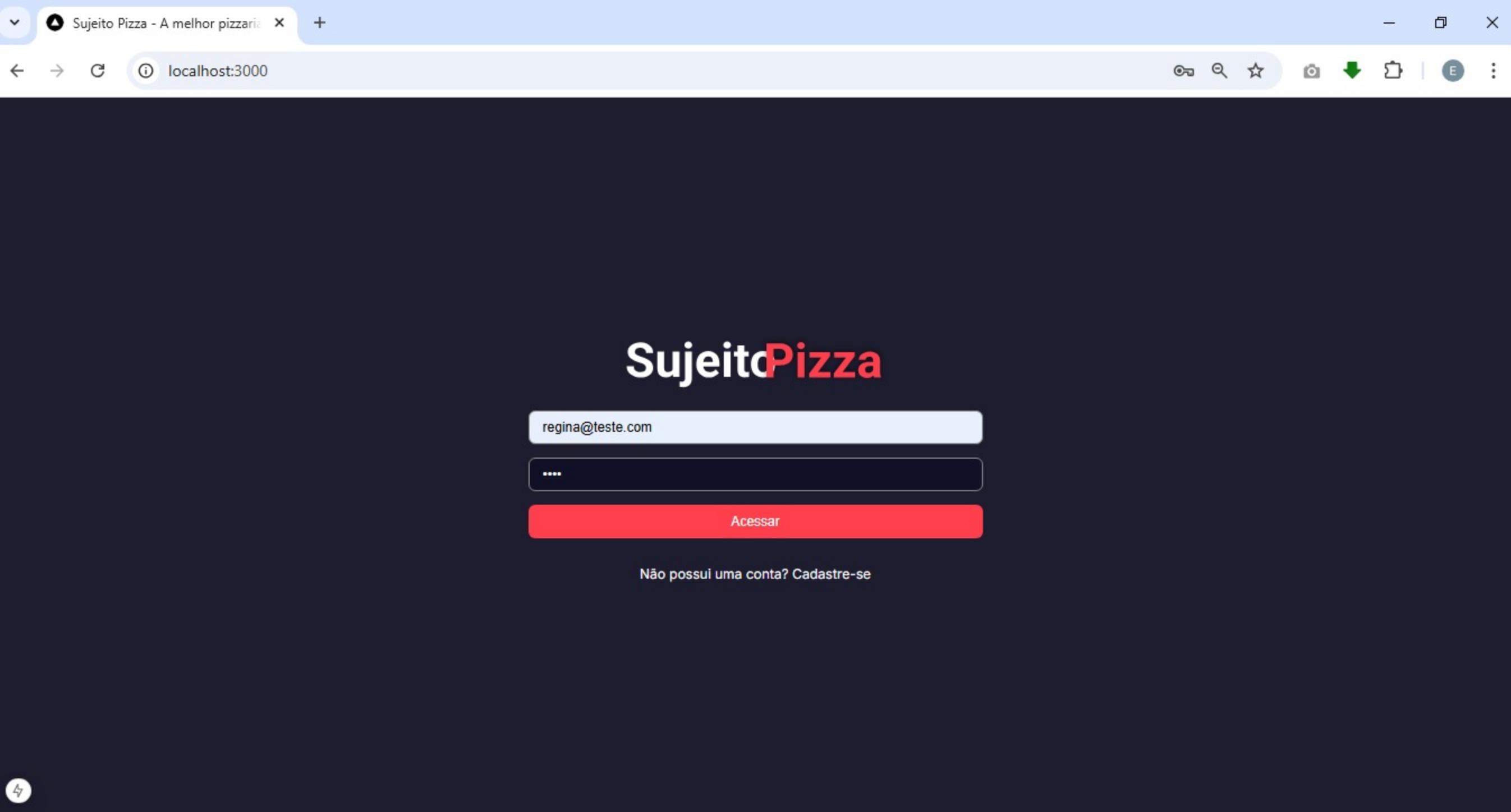
Já possui uma conta? [Faça o login](#)



SujeitoPizza

Digite seu email...

Não possui uma conta? [Cadastre-se](#)



Sujeito Pizza - A melhor pizzaria

localhost:3000

regina@teste.com

....

Acessar

Não possui uma conta? Cadastre-se

SujeitoPizza

[Categoria](#) [Produto](#) [→]

Últimos pedidos



Mesa 35

Mesa 55



Pedidos feitos

Últimos pedidos

Mesa 10

Mesa 11



Detalhes do pedido

Mesa 35**2 - Burger Triplo Dev**

Hambúrguer Triplo com Cheddar

1 - Dev Burger Salada

Burgão Dev de Salada com 1 hambúrguer de 160g

Concluir pedido

SujeitoPizza

Categoria Produto [→]

Últimos pedidos

Mesa 35

Mesa 55



Pedidos

Últimos pedidos

Mesa 55

Mesa 56



Detalhes do pedido

Mesa 55**2 - Coca lata**

Coca-Cola lata 350ml

3 - Pizza de calabresa

Pizza de calabresa saborosa

Concluir pedido

SujeitoPizza

Categoria Produto [→]

Últimos pedidos

Mesa 35

Mesa 55

UC: Projetos

Assunto:

Projeto Pizzaria – Frontend_02 (NextJS)

Sistema Pizzaria Web

64 – Finalizando Projeto Web



SujeitoPizza

Categoria Produto [→]

Últimos pedidos

Mesa 35

Mesa 55

Novo produto

**bebidas**

Digite o nome do produto...

Preço do produto...

Digite a descrição do produto...

Cadastrar produto

Abrir

Área de Trabalho > arquivos

Pesquisar em arquivos

Organizar Nova pasta

acai

açai

hamburger duplo

hamburger triplo

hamburger

sorvete 1

Nome: acai Arquivos personalizados

Abrir Cancelar

Digite o nome do produto...

Preço do produto...

Digite a descrição do produto...

Cadastrar produto

Categoria Produto [→]

Novo produto



bebidas

Digite o nome do produto...

Preço do produto...

Digite a descrição do produto...

Cadastrar produto

Novo produto



bebidas

bebidas

pizzas

sobremesas

espeto

hamburger

acompanhamentos

sorvete

esfihas

Cadastrar produto

Novo produto



bebidas

bebidas

pizzas

sobremesas

espeto

hamburger

acompanhamentos

sorvete

esfihas

Cadastrar produto

Novo produto



sobremesas

Açaí 500ml

30

pote de açaí de 500ml

Cadastrar produto

SujeitoPizza

[Categoria](#) [Produto](#) [\[→\]](#)

Últimos pedidos

Mesa 35

Mesa 55

File Edit View Ultimate Tools Help localhost:5432/pizzaria - Beekeeper Studio

pizzaria C +

Query #1 products [all] orders [all]

Filter

ENTITIES 217 +

public

- _prisma_migrations
- categories
- items
- orders
- products
- users

information_schema

pg_catalog

id equals Enter Value

| | id | name | price | description | banner | created_at | update_at | category |
|---|---------------------------------------|-----------------------|-------|---------------------|-----------------|----------------|-------------------------|----------|
| 1 | 0cd97aa3-1583-458e-b0ef-549fd1f40e37 | Burger Completo Duplo | 45 | Hambuger Dupl... | 040cfab507... | 2024-11-29 ... | 2024-11-29 01:57:17.362 | 1e4ffc |
| 2 | 1cd5c28d-6392-4d30-9b2d-91278fccef7f0 | Coca-Cola 2L | 10 | Coca-Cola gelad... | b98497f70a... | 2024-11-25 ... | 2024-11-25 23:47:24.095 | 00d8f7 |
| 3 | 6ad3ded6-115b-468c-89eb-25de1a9459... | Burger Triplo Dev | 50 | Hamburger Tripl... | 9b5565412c... | 2024-11-29 ... | 2024-11-29 02:14:03.556 | 1e4ffc |
| 4 | 7f66d79a-ceb2-4175-b4a6-dc5208a9f50e | Pizza de calabresa | 46 | Pizza de calabre... | 8a652b025a... | 2024-11-25 ... | 2024-11-25 22:06:31.016 | e2142d |
| 5 | 96eb4477-824b-4137-8251-dbce1d72f7... | Dev Burger Salada | 40 | Burgão Dev de S... | c83f6cc7a8b... | 2024-11-29 ... | 2024-11-29 01:06:24.05 | 1e4ffc |
| 6 | ca5c64fa-e4dd-4b67-9c33-2c4961d4b0... | Coca lata | 5 | Coca-Cola lata 3... | 765f54ffa7f0... | 2024-11-25 ... | 2024-11-25 22:17:10.368 | 00d8f7 |
| 7 | dc850cd8-a417-4719-9494-ddff3f53b8ed | Açaí 300ml | 12 | Pote de Açaí 30... | d3c76275c1... | 2024-11-29 ... | 2024-11-29 01:29:50.485 | f44fb5 |

pizzaria

Query #1 products [all] orders [all]

ENTITIES 217

public

- _prisma_migrations
- categories
- items
- orders
- products
- users

information_schema

pg_catalog

| | id | equals | Enter Value | |
|---|---------------------------------------|-----------------------|-------------|---------------------|
| 1 | 0cd97aa3-1583-458e-b0ef-549fd1f40e37 | Burger Completo Duplo | 45 | Hambuger Dupl... |
| 2 | 1cd5c28d-6392-4d30-9b2d-91278fccef7f0 | Coca-Cola 2L | 10 | Coca-Cola gelad... |
| 3 | 37e1bb99-2313-4a82-98c3-769d93409a... | Açaí 500ml | 30 | pote de açaí de ... |
| 4 | 6ad3ded6-115b-468c-89eb-25de1a9459... | Burger Triplo Dev | 50 | Hamburger Tripl... |
| 5 | 7f66d79a-ceb2-4175-b4a6-dc5208a9f50e | Pizza de calabresa | 46 | Pizza de calabre... |
| 6 | 96eb4477-824b-4137-8251-dbce1d72f7... | Dev Burger Salada | 40 | Burgão Dev de S... |
| 7 | ca5c64fa-e4dd-4b67-9c33-2c4961d4b0... | Coca lata | 5 | Coca-Cola lata 3... |
| 8 | dc850cd8-a417-4719-9494-ddff3f53b8ed | Açaí 300ml | 12 | Pote de Açaí 30... |

Refresh Table (ctrl+r or F5)

pizzaria

Query #1 products [all] orders [all]

Filter

ENTITIES 217

public

- _prisma_migrations
- categories
- items
- orders
- products
- users

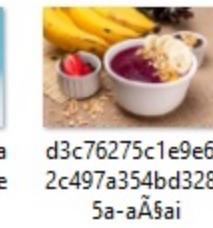
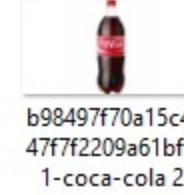
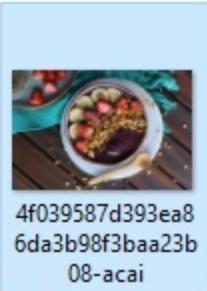
information_schema

pg_catalog

| | id | equals | Enter Value | | | | | | | | | | | | | | | | | |
|---|---------------------------------------|-----------------------|-------------|---------------------|-----------------|----------------|-------------------------|--------|--|--|--|--|--|--|--|--|--|--|--|--|
| 1 | 0cd97aa3-1583-458e-b0ef-549fd1f40e37 | Burger Completo Duplo | 45 | Hambuger Dupl... | 040cfab507... | 2024-11-29 ... | 2024-11-29 01:57:17.362 | 1e4ffd | | | | | | | | | | | | |
| 2 | 1cd5c28d-6392-4d30-9b2d-91278fce7f0 | Coca-Cola 2L | 10 | Coca-Cola gelad... | b98497f70a... | 2024-11-25 ... | 2024-11-25 23:47:24.095 | 00d8f | | | | | | | | | | | | |
| 3 | 37e1bb99-2313-4a82-98c3-769d93409a... | Açaí 500ml | 30 | pote de açaí de ... | 4f039587d3... | 2024-12-05 ... | 2024-12-05 00:34:31.367 | 80de4 | | | | | | | | | | | | |
| 4 | 6ad3ded6-115b-468c-89eb-25de1a9459... | Burger Triplo Dev | 50 | Hamburger Tripl... | 9b5565412c... | 2024-11-29 ... | 2024-11-29 02:14:03.556 | 1e4ffd | | | | | | | | | | | | |
| 5 | 7f66d79a-ceb2-4175-b4a6-dc5208a9f50e | Pizza de calabresa | 46 | Pizza de calabre... | 8a652b025a... | 2024-11-25 ... | 2024-11-25 22:06:31.016 | e2142 | | | | | | | | | | | | |
| 6 | 96eb4477-824b-4137-8251-dbce1d72f7... | Dev Burger Salada | 40 | Burgão Dev de S... | c83f6cc7a8b... | 2024-11-29 ... | 2024-11-29 01:06:24.05 | 1e4ffd | | | | | | | | | | | | |
| 7 | ca5c64fa-e4dd-4b67-9c33-2c4961d4b0... | Coca lata | 5 | Coca-Cola lata 3... | 765f54ffa7f0... | 2024-11-25 ... | 2024-11-25 22:17:10.368 | 00d8f | | | | | | | | | | | | |
| 8 | dc850cd8-a417-4719-9494-ddff3f53b8ed | Açaí 300ml | 12 | Pote de Açaí 30... | d3c76275c1... | 2024-11-29 ... | 2024-11-29 01:29:50.485 | f44fb5 | | | | | | | | | | | | |

| Nome | Tipo | Data de modificação | Tamanho |
|--|-------------------|---------------------|---------|
|  backend | Pasta de arquivos | 21/11/2024 23:02 | |
|  frontend | Pasta de arquivos | 26/11/2024 15:58 | |

| Este Computador > Área de Trabalho > Pizzaria > backend | | | | |
|---|--|---------------------|-----------------------|---------|
| Nome | | Data de modificação | Tipo | Tamanho |
|  node_modules | | 21/11/2024 22:18 | Pasta de arquivos | |
|  prisma | | 20/11/2024 09:58 | Pasta de arquivos | |
|  src | | 21/11/2024 22:19 | Pasta de arquivos | |
|  tmp | | 04/12/2024 21:34 | Pasta de arquivos | |
|  tmp | | 21/11/2024 22:32 | Pasta de arquivos | |
|  .env | | 20/11/2024 18:15 | Arquivo ENV | 1 KB |
|  .gitignore | | 20/11/2024 08:55 | Arquivo Fonte Git ... | 1 KB |
|  package | | 21/11/2024 22:16 | Arquivo Fonte JSON | 1 KB |
|  tsconfig | | 21/11/2024 19:00 | Arquivo Fonte JSON | 13 KB |
|  yarn.lock | | 21/11/2024 22:16 | Arquivo LOCK | 187 KB |
|  yarn-error | | 16/11/2024 18:30 | Documento de Te... | 2 KB |





Pesquise no Google ou digite um URL



Google



Web Store



Adicionar ata...

pizzaria

Query #1 products [all] orders [all]

Filter

ENTITIES 217

| | id | name | price | description | banner | created_at... | update_at | category |
|---|---------------------------------------|-----------------------|-------|---------------------|---------------|-------------------------|-------------------------|----------|
| 1 | 0cd97aa3-1583-458e-b0ef-549fd1f40e37 | Burger Completo Duplo | 45 | Hambuger Dupl... | 040cfab507... | 2024-11-29 ... | 2024-11-29 01:57:17.362 | 1e4ffc |
| 2 | 1cd5c28d-6392-4d30-9b2d-91278fce7f0 | Coca-Cola 2L | 10 | Coca-Cola gelad... | b98497f70a... | 2024-11-25 ... | 2024-11-25 23:47:24.095 | 00d8f7 |
| 3 | 37e1bb99-2313-4a82-98c3-769d93409a... | Açaí 500ml | 30 | pote de açaí de ... | 4f039587d? | 2024-12-05 ... | 2024-12-05 00:34:31.367 | 80de43 |
| 4 | 6ad3ded6-115b-468c-89eb-25de1a9459... | Burger Triplo Dev | 50 | Hamburger Tripl... | 9b5565412 | 2024-11-29 02:14:03.556 | 2024-11-29 02:14:03.556 | 1e4ffc |
| 5 | 7f66d79a-ceb2-4175-b4a6-dc5208a9f50e | Pizza de calabresa | 46 | Pizza de calabre... | 8a652b025 | 2024-11-25 22:06:31.016 | 2024-11-25 22:06:31.016 | e21423 |
| 6 | 96eb4477-824b-4137-8251-dbce1d72f7... | Dev Burger Salada | 40 | Burgão Dev de S... | c83f6cc7a | 2024-11-29 01:06:24.05 | 2024-11-29 01:06:24.05 | 1e4ffc |
| 7 | ca5c64fa-e4dd-4b67-9c33-2c4961d4b0... | Coca lata | 5 | Coca-Cola lata 3... | 765f54ffa7 | 2024-11-25 22:17:10.368 | 2024-11-25 22:17:10.368 | 00d8f7 |
| 8 | dc850cd8-a417-4719-9494-ddff3f53b8ed | Açaí 300ml | 12 | Pote de Açaí 30... | d3c76275c | 2024-11-29 01:29:50.485 | 2024-11-29 01:29:50.485 | f44fb5 |

id equals Enter Value

4f039587d? Edit in modal Shift+Enter

Set as NULL

Copy Ctrl+C

Copy as TSV for Excel

Copy as JSON

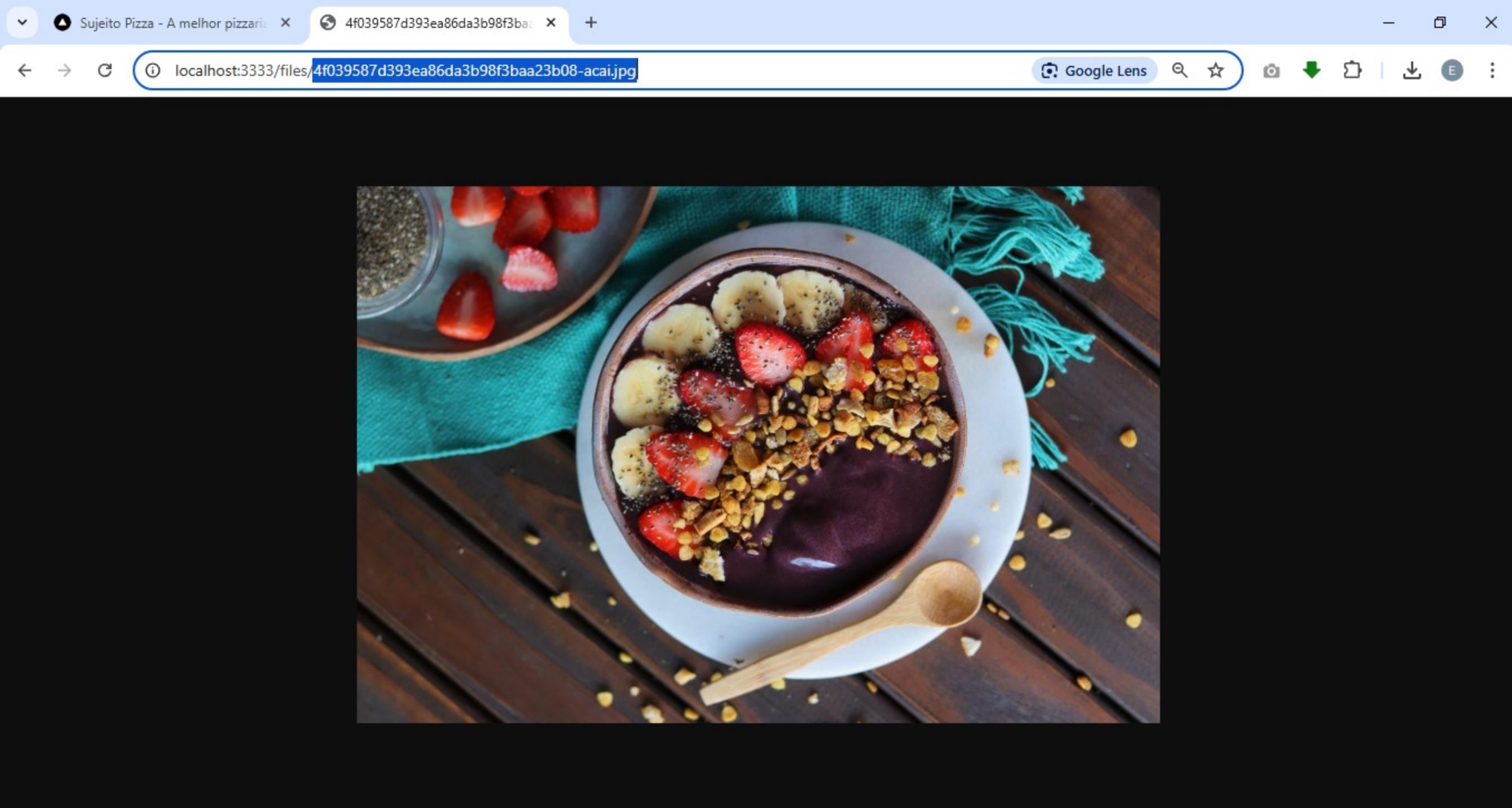
Copy as Markdown

Copy as SQL

Paste Ctrl+V

Clone row Ctrl+D

Delete row Delete



frontend

EXPLORER

FRONTEND

- > .next
- > node_modules
- > public
- src
 - app
 - dashboard
 - > category
 - components
 - > button
 - > header
 - modal
 - > index.tsx
 - > styles.module.scss
 - > orders
 - > product
 - layout.tsx
 - page.tsx
 - > fonts
 - > signup
 - favicon.ico
 - > globals.scss
 - layout.tsx
 - > page.module.scss
 - page.tsx
 - > lib
 - > providers
- > OUTLINE
- > TIMELINE

index.tsx ...\\orders

index.tsx ...\\modal

```
src > app > dashboard > components > modal > index.tsx > Modalorder
  32   <b>{order[0].order.name}</b>
  33   </span>
  34 }
  35
  36   {order.map( item => (
  37     <section className={styles.item} key={item.id}>
  38       <span>{item.amount} - <b>{item.product.name}</b></span>
  39       <span className={styles.description}>
  40         {item.product.description}</span>
  41       </section>
  42     ))}
  43
  44   <img
  45     src={`${http://localhost:3333/files/${order[0].product.
  46       banner}}`}
  47     width={300}
  48     height={300}
  49   />
  50
  51   <button className={styles.buttonOrder} onClick=
  52     {handleFinishOrder}>
  53     Concluir pedido
  54   </button>
  55
  56 </article>
  57
  58 </section>
  59
  60 }
```

Ln 44, Col 1 (157 selected) Spaces: 2 CRLF TypeScript JSX Prettier

Sujeito Pizza

Categoria Produto [→]

Últimos pedidos

- Mesa 35
- Mesa 55

frontend

EXPLORER

FRONTEND

- > .next
- > node_modules
- > public
- src
 - app
 - dashboard
 - > category
 - > components
 - > button
 - > header
 - > modal
 - index.tsx
 - styles.module.scss
 - > orders
 - > product
 - layout.tsx
 - page.tsx
 - > fonts
 - > signup
 - favicon.ico
 - globals.scss
 - layout.tsx
 - page.module.scss
 - page.tsx
 - > lib
 - > providers
- > OUTLINE
- > TIMELINE

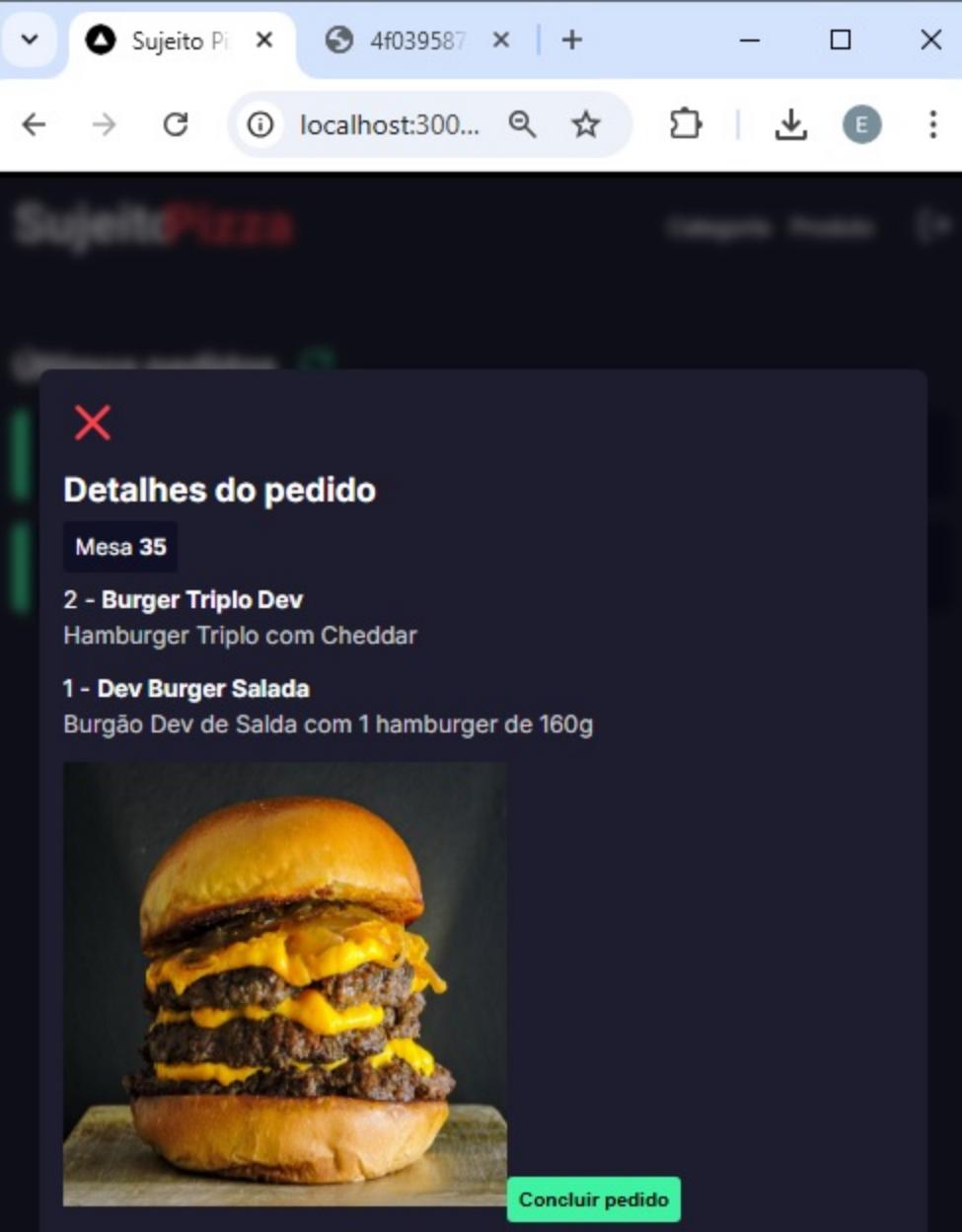
index.tsx ...\\orders index.tsx ...\\modal

```
src > app > dashboard > components > modal > index.tsx > Modalorder
  32     <b>{order[0].order.name}</b>
  33   </span>
  34 }
  35
  36   {order.map( item => (
  37     <section className={styles.item} key={item.id}>
  38       <span>{item.amount} - <b>{item.product.name}</b></span>
  39       <span className={styles.description}>
  40         {item.product.description}</span>
  41     </section>
  42   ))}

  44 <img
  45   src={`http://localhost:3333/files/${order[0].product.
  46     banner}`}
  47   width={300}
  48   height={300}
  49 >

  50   <button className={styles.buttonOrder} onClick=
  51     {handleFinishOrder}>
  52     Concluir pedido
  53   </button>
  54 </article>
  55
  56 </section>
  57
  58 </dialog>
  59 )
  60 }
```

Ln 44, Col 1 (157 selected) Spaces: 2 CRLF ⓘ TypeScript JSX ✅ Prettier



frontend

index.tsx ...\\orders index.tsx ...\\modal

```
src > app > dashboard > components > modal > index.tsx > Modalorder
  32   <b>{order[0].order.name}</b>
  33   </span>
  34 }
  35
  36   {order.map( item => (
  37     <section className={styles.item} key={item.id}>
  38       <span>{item.amount} - <b>{item.product.name}</b></span>
  39       <span className={styles.description}>
  40         {item.product.description}</span>
  41     </section>
  42   ))}

  44   <img
  45     src={`${http://localhost:3333/files/${order[0].product.
  46       banner}}`}
  47     width={300}
  48     height={300}
  49   />

  50   <button className={styles.buttonOrder} onClick=
  51     {handleFinishOrder}>
  52     Concluir pedido
  53   </button>
  54 </article>
  55 </section>
  56 </dialog>
  57 )
  58 )
  59 )
  60 }
```

Ln 44, Col 1 (157 selected) Spaces: 2 CRLF TypeScript JSX Prettier

Sujeito Pi 4f039587 localhost:300... E

Detalhes do pedido

Mesa 55

2 - Coca lata

Coca-Cola lata 350ml

3 - Pizza de calabresa

Pizza de calabresa saborosa



Concluir pedido

frontend

index.tsx ...\\orders index.tsx ...\\modal

```
src > app > dashboard > components > modal > index.tsx > Modalorder > order.map() callback
  32           <b>{order[0].order.name}</b>
  33           </span>
  34       )
  35
  36       {order.map( item => (
  37         <section className={styles.item} key={item.id}>
  38           <img ...
  39             src={^http://localhost:3333/files/${order[0].
  40               product.banner}`}
  41             width={300}
  42             height={300}
  43           />
  44           <span>{item.amount} - <b>{item.product.name}</b></span>
  45           <span className={styles.description}>
  46             {item.product.description}</span>
  47         </section>
  48     )));
  49
  50       <button className={styles.buttonOrder} onClick=
  51         {handleFinishOrder}>
  52           Concluir pedido
  53         </button>
  54
  55     </article>
  56
  57   </section>
  58
  59 )
```

Ln 38, Col 1 (177 selected) Spaces: 2 CRLF TypeScript JSX Prettier

Detalhes do pedido

Mesa 55

2 - Coca lata

Coca-Cola lata 350ml

3 - Pizza de calabresa

Pizza de calabresa saborosa

frontend

index.tsx ...\\orders index.tsx ...\\modal

```
src > app > dashboard > components > modal > index.tsx > Modalorder > order.map() callback
  32
  33           <b>{order[0].order.name}</b>
  34       </span>
  35   )
  36   {order.map( item => (
  37     <section className={styles.item} key={item.id}>
  38       <img
  39         src={`http://localhost:3333/files/${order[0].product.banner}`}
  40         width={300}
  41         height={300}
  42       />
  43       <span>{item.amount} - <b>{item.product.name}</b></span>
  44       <span className={styles.description}>
  45         {item.product.description}</span>
  46     </section>
  47   )))
  48
  49   <button className={styles.buttonOrder} onClick={handleFinishOrder}>
  50     Concluir pedido
  51   </button>
  52
  53 </article>
  54
  55 </section>
  56
  57 </dialog>
  58
  59 )
```

Ln 39, Col 76 (23 selected) Spaces: 2 UTF-8 CRLF TypeScript JSX ✓ Prettier

Sujeito Pi 4f039587

localhost:300...

X Detalhes do pedido Mesa 55

 SABOR ORIGINAL
Coca-Cola
DESENTE 1886
350 ml

2 - Coca lata
Coca-Cola lata 350ml

 SABOR ORIGINAL
Coca-Cola
DESENTE 1886
350 ml

3 - Pizza de calabresa
Pizza de calabresa saborosa

frontend

EXPLORER

FRONTEND

- > .next
- > node_modules
- > public
- src
 - app
 - dashboard
 - > category
 - components
 - > button
 - > header
 - modal
 - index.tsx
 - styles.module.scss
 - > orders
 - > product
 - layout.tsx
 - page.tsx
 - > fonts
 - > signup
 - favicon.ico
 - globals.scss
 - layout.tsx
 - page.module.scss
 - page.tsx
 - > lib
 - > providers

OUTLINE

TIMELINE

index.tsx ...\\orders

index.tsx ...\\modal

```
rc > app > dashboard > components > modal > index.tsx > Modalorder > order.map() callback
  32   <b>{order[0].order.name}</b>
  33   </span>
  34 }
  35
  36 {order.map( item => (
  37   <section className={styles.item} key={item.id}>
  38     <img
  39       src={`http://localhost:3333/files/${item.product.
  banner}`}
  40       width={300}
  41       height={300}
  42     />
  43     <span>{item.amount} - <b>{item.product.name}</b></span>
  44     <span className={styles.description}>
  45       {item.product.description}</span>
  46     </section>
  47   ))}

  48   <button className={styles.buttonOrder} onClick=
  49     {handleFinishOrder}>
  50       Concluir pedido
  51     </button>
  52   </article>
  53 </section>
  54 </dialog>
  55 )
  56 }
  57 )
  58 )
  59 }
```

Ln 39, Col 72 (19 selected) Spaces: 2 CRLF TypeScript JSX Prettier

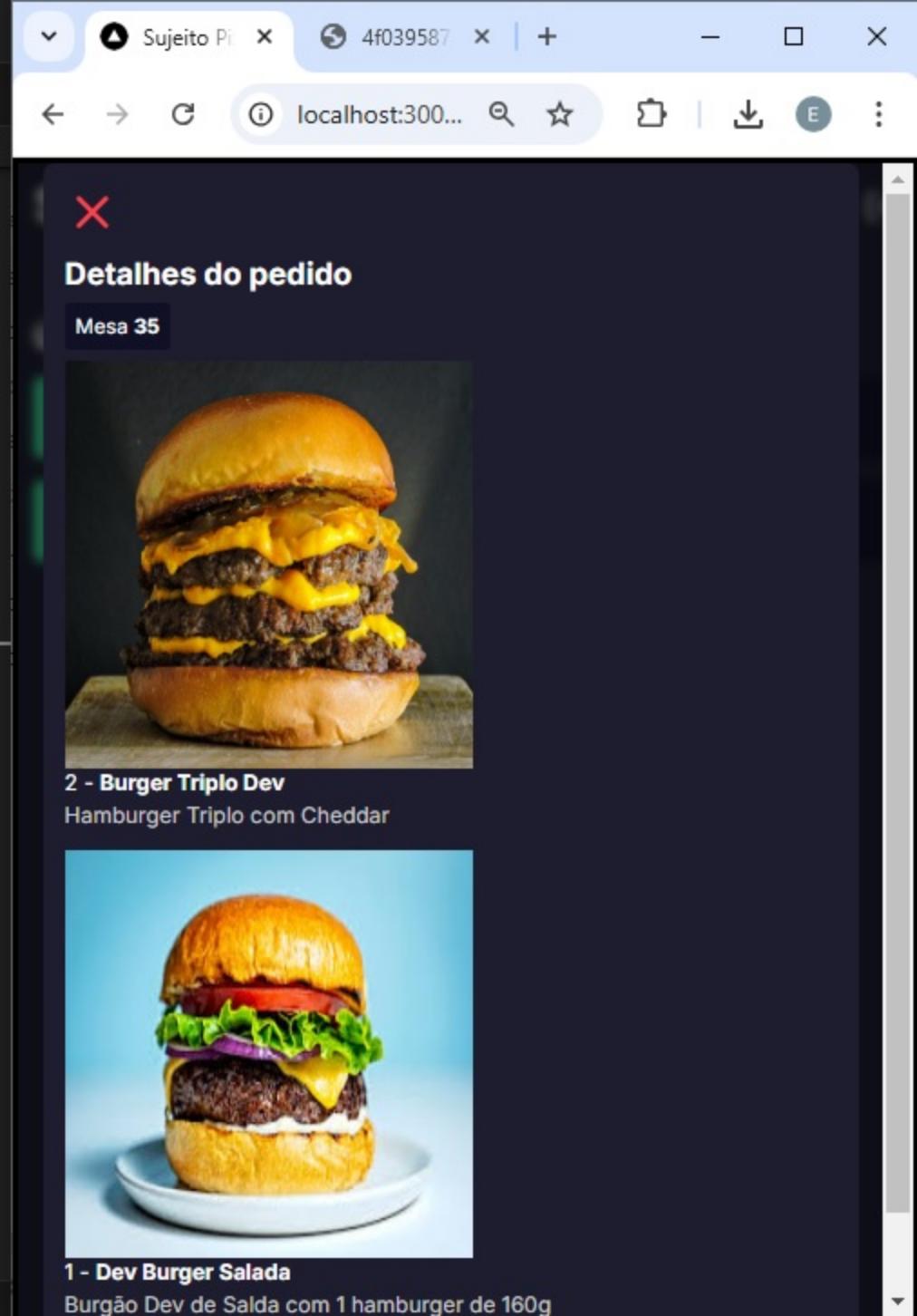
SujeitoPizza

Categoria Produto [→]

Últimos pedidos

Mesa 35

Mesa 55



A screenshot of a web browser displaying a burger order details page. The page has a dark header with a red 'X' icon and the text 'Detalhes do pedido'. Below the header, it says 'Mesa 35' and shows two images of burgers. The top burger is labeled '2 - Burger Triplo Dev' and 'Hambúrguer Triplo com Cheddar'. The bottom burger is labeled '1 - Dev Burger Salada' and 'Burgão Dev de Salada com 1 hambúrguer de 160g'.

index.tsx ...\\orders

index.tsx ...\\modal

```
rc > app > dashboard > components > modal > index.tsx > Modalorder > order.map() callback
```

32 {order[0].order.name}

33

34)

35

36 {order.map(item => (

37 <section className={styles.item} key={item.id}>

38 <img

39 src={^http://localhost:3333/files/\${item.product.
banner}}^}

40 width={300}

41 height={300}

42 />

43 {item.amount} - {item.product.name}

44

45 {item.product.description}

46 </section>

47))}

48

49 <button className={styles.buttonOrder} onClick={handleFinishOrder}>

50 Concluir pedido

51 </button>

52

53 </article>

54

55 </section>

56

57 </dialog>

58

59 }

Ln 39, Col 72 (19 selected) Spaces: 2 UTF-8 CRLF TypeScript JSX Prettier

EXPLORER

FRONTEND

- .next
- node_modules
- public
- src
 - app
 - dashboard
 - category
 - components
 - button
 - header
 - modal
 - index.tsx
 - styles.module.scss
 - orders
 - product
 - layout.tsx
 - page.tsx
 - fonts
 - signup
 - favicon.ico
 - globals.scss
 - layout.tsx
 - page.module.scss
 - page.tsx
 - lib
 - providers
- OUTLINE
- TIMELINE

frontend

index.tsx ...\\orders index.tsx ...\\modal

```
rc > app > dashboard > components > modal > index.tsx > Modalorder > order.map() callback
  32           <b>{order[0].order.name}</b>
  33           </span>
  34       )
  35
  36   {order.map( item => (
  37     <section className={styles.item} key={item.id}>
  38       <img
  39         src={^http://localhost:3333/files/${item.product.
  40         banner}^}
  41         width={300}
  42         height={300}
  43       />
  44       <span>{item.amount} - <b>{item.product.name}</b></span>
  45       <span className={styles.description}>
  46         {item.product.description}</span>
  47     </section>
  48   ))}
  49
  50   <button className={styles.buttonOrder} onClick=
  51     {handleFinishOrder}>
  52     Concluir pedido
  53   </button>
  54
  55 </article>
  56
  57 </section>
  58
  59 )}
```

localhost:300... 4f039587

Detalhes do pedido

Mesa 35



2 - Burger Triplo Dev
Hamburger Triplo com Cheddar



1 - Dev Burger Salada
Burgão Dev de Salada com 1 hambúrguer de 160g

EXPLORER

FRONTEND

- .next
- node_modules
- public
- src
 - app
 - category
 - components
 - button
 - header
 - modal
 - index.tsx
 - styles.module.scss
 - orders
 - product
 - layout.tsx
 - page.tsx
 - fonts
 - signup
 - favicon.ico
 - globals.scss
 - layout.tsx
 - page.module.scss
 - page.tsx
 - lib
 - providers
- OUTLINE
- TIMELINE

2 selections (6 characters selected) Spaces: 2 UTF-8 CRLF TypeScript JSX Prettier

frontend

index.tsx ...\\orders index.tsx ...\\modal

```
rc > app > dashboard > components > modal > index.tsx > Modalorder > order.map() callback
  32   <b>{order[0].order.name}</b>
  33   </span>
  34 }
  35
  36 {order.map( item => (
  37   <section className={styles.item} key={item.id}>
  38     <img
  39       src={`http://localhost:3333/files/${item.product.
  40         banner}`}
  41       width={80}
  42       height={80}
  43     />
  44     <span>{item.amount} - <b>{item.product.name}</b></span>
  45     <span className={styles.description}>
  46       {item.product.description}</span>
  47     </section>
  48   )))
  49
  50   <button className={styles.buttonOrder} onClick=
  51     {handleFinishOrder}>
  52     Concluir pedido
  53   </button>
  54
  55 </article>
  56
  57 </section>
  58
  59 )}
```

localhost:300...

Sujeito Pi 4f039587

Detalhes do pedido

Mesa 35

2 - Burger Triplo Dev

Hamburger Triplo com Cheddar

1 - Dev Burger Salada

Burgão Dev de Salada com 1 hambúrguer de 160g

Concluir pedido

frontend

EXPLORER

FRONTEND

- > .next
- > node_modules
- > public
- src
 - app
 - dashboard
 - > category
 - components
 - > button
 - > header
 - modal
 - index.tsx
 - styles.module.scss
 - > orders
 - > product
 - layout.tsx
 - page.tsx
 - > fonts
 - > signup
 - favicon.ico
 - globals.scss
 - layout.tsx
 - page.module.scss
 - page.tsx
 - > lib
 - > providers
- > OUTLINE
- > TIMELINE

index.tsx ...\\orders

index.tsx ...\\modal

```
rc > app > dashboard > components > modal > index.tsx > Modalorder > order.map() callback
  32   <b>{order[0].order.name}</b>
  33   </span>
  34 }
  35
  36 {order.map( item => (
  37   <section className={styles.item} key={item.id}>
  38     <img
  39       src={`http://localhost:3333/files/${item.product.
  40         banner}`}
  41       width={80}
  42       height={80}
  43     />
  44     <span>{item.amount} - <b>{item.product.name}</b></span>
  45     <span className={styles.description}>
  46       {item.product.description}</span>
  47     </section>
  48   ))}

  49   <button className={styles.buttonOrder} onClick=
  50     {handleFinishOrder}>
  51       Concluir pedido
  52     </button>
  53   </article>
  54 </section>
  55 </dialog>
  56 )
  57 )
  58 )
  59 }
```

2 selections (9 characters selected) Spaces: 2 CRLF TypeScript JSX Prettier

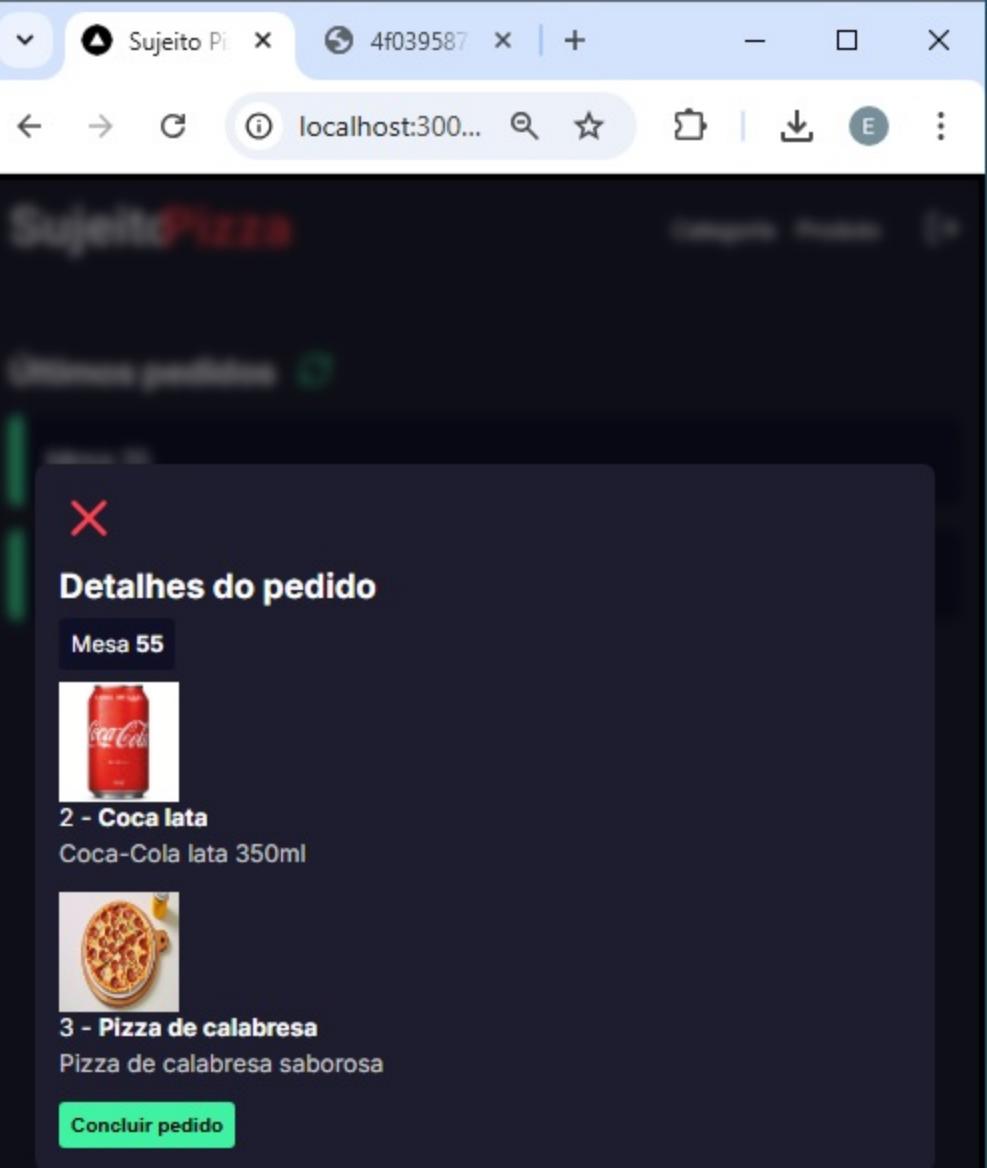
SujeitoPizza

Categoria Produto [→]

Últimos pedidos

Mesa 35

Mesa 55



frontend

EXPLORER

FRONTEND

- > .next
- > node_modules
- > public
- src
 - app
 - dashboard
 - > category
 - > components
 - button
 - header
 - modal
 - index.tsx
 - styles.module.scss

index.tsx ... \orders

index.tsx ... \modal

```
rc > app > dashboard > components > modal > index.tsx > Modalorder > order.map() callback
  32   <b>{order[0].order.name}</b>
  33   </span>
  34 }
  35
  36 {order.map( item => (
  37   <section className={styles.item} key={item.id}>
  38     <img
  39       src={^http://localhost:3333/files/${item.product.
  40         banner}^}
  41       width={80}
  42       height={80}
  43     />
  44     <span>{item.amount} - <b>{item.product.name}</b></span>
  45     <span className={styles.description}>
  46       {item.product.description}</span>
  47     </section>
  48   ))
  49
  50   <button className={styles.buttonOrder} onClick=
  51     {handleFinishOrder}>
  52     Concluir pedido
  53   </button>
  54
  55 </article>
  56
  57 </section>
  58
  59 )}
```

2 selections (9 characters selected) Spaces: 2 UTF-8 CRLF TypeScript JSX Prettier

frontend

index.tsx ...\\orders index.tsx ...\\modal

```
rc > app > dashboard > components > modal > index.tsx > Modalorder > order.map() callback
  32           <b>{order[0].order.name}</b>
  33           </span>
  34       )
  35
  36       {order.map( item => (
  37         <section className={styles.item} key={item.id}>
  38           <img ...
  39             src={`http://localhost:3333/files/${item.product.
  40               banner}`}
  41             width={80}
  42             height={80}
  43           />
  44           <span>{item.amount} - <b>{item.product.name}</b></span>
  45           <span className={styles.description}>
  46             {item.product.description}</span>
  47         </section>
  48     )))
  49
  50         <button className={styles.buttonOrder} onClick=
  51           {handleFinishOrder}>
  52             Concluir pedido
  53         </button>
  54
  55     </article>
  56
  57   </section>
  58
  59 )}
```

Ln 38, Col 1 (171 selected) Spaces: 2 CRLF TypeScript JSX Prettier

SujeitoPi x 4f039587 x +

localhost:300... 🔍 ⚡ E :

SujeitoPizza

Categoria Produto [→]

Últimos pedidos ↻

- Mesa 35
- Mesa 55

frontend

index.tsx ...\\orders index.tsx ...\\modal

```
rc > app > dashboard > components > modal > index.tsx > Modalorder > order.map() callback
  32           <b>{order[0].order.name}</b>
  33           </span>
  34       )
  35
  36       {order.map( item => (
  37           <section className={styles.item} key={item.id}>
  38           <span>{item.amount} - <b>{item.product.name}</b></span>
  39           <span className={styles.description}>
  40             {item.product.description}</span>
  41         </section>
  42     )))
  43
  44     <button className={styles.buttonOrder} onClick={handleFinishOrder}>
  45       Concluir pedido
  46     </button>
  47   </article>
  48 </section>
  49 </dialog>
  50 )
  51
  52 )
  53
  54 }
```

Ln 37, Col 1 (265 selected) Spaces: 2 UTF-8 CRLF TypeScript JSX Prettier

Sujeito Pizza

Categoria Produto [→]

Últimos pedidos

- Mesa 35
- Mesa 55

The screenshot shows a developer's environment with multiple windows open:

- VS Code Editor:** The main window displays the file `index.tsx` from the `modal` directory. The code is a modal component for viewing order details. A yellow circular cursor is positioned on line 37, which contains the opening brace of a map function. The code uses TypeScript and JSX.
- Browser Preview:** An integrated browser window shows the rendered modal component. It has a dark theme with a red header bar. The modal title is "Detalhes do pedido" (Order Details). Inside, it shows an order for "Mesa 35" (Table 35) with two items:
 - 2 - Burger Triplo Dev: Hamburger Triplo com Cheddar
 - 1 - Dev Burger Salada: Burgão Dev de Salada com 1 hambúrguer de 160gA green button at the bottom right says "Concluir pedido" (Finish Order).
- Terminal:** The bottom-left corner shows a terminal window with the command `npm start` and its output, indicating the application is running on `localhost:3001`.

The image shows a developer's workspace with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "index.tsx" is open in the editor.
- Content:** The code is a TypeScript component for a modal window. It includes logic to render a table for a specific order and a map function to render individual order items. A button at the bottom is used to handle the completion of the order.

```
index.tsx ...\\orders
src > app > dashboard > components > modal > index.tsx > Modalorder > order.map() callback
26   <span className={styles.table}>
27     | Mesa <b>{order[0].order.table}</b>
28   </span>
29
30   {order[0].order?.name && (
31     <span className={styles.name}>
32       <b>{order[0].order.name}</b>
33     </span>
34   )}
35
36   {order.map( item => (
37     <section className={styles.item} key={item.id}>
38       <span>{item.amount} - <b>{item.product.name}</b></span>
39       <span className={styles.description}>
40         {item.product.description}</span>
41     </section>
42   )));
43
44   <button className={styles.buttonOrder} onClick={handleFinishOrder}>
45     Concluir pedido
46   </button>
47
48   </article>
49
50 </section>
51
52 </dialog>
53 )
54 }
```

Browser Preview:

- Title:** Sujeto Pi
- URL:** localhost:300...
- Content:** A modal window titled "Detalhes do pedido" (Details of the order) is displayed. It shows the table for order number 35, listing two items: "2 - Burger Triplo Dev" (Hamburger Triplo com Cheddar) and "1 - Dev Burger Salada" (Burgão Dev de Salada com 1 hamburger de 160g). A green "Concluir pedido" (Finish order) button is visible at the bottom right.

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under `FRONTEND`. The `modal` folder is currently selected.
- File:** `index.tsx` is open in the editor.
- Content:** The code is a modal component for viewing order details. It includes logic to check if an order has a name and displays it. It also maps over an array of items to create sections for each item with its quantity and description. A button at the bottom allows finishing the order.

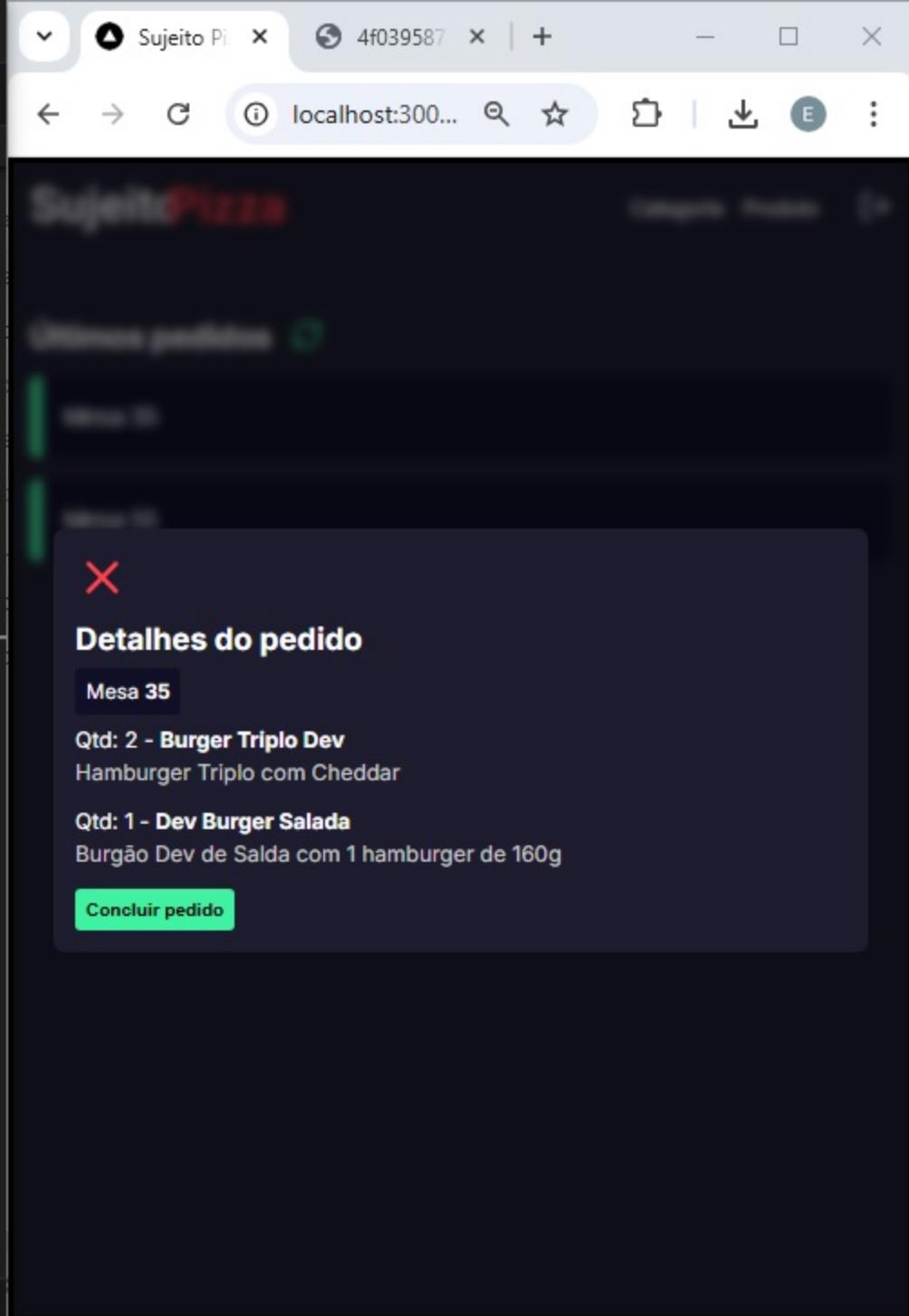
```
rc > app > dashboard > components > modal > index.tsx > Modalorder > order.map() callback
  26   <span className={styles.table}>
  27     | Mesa <b>{order[0].order.table}</b>
  28   </span>
  29
  30   {order[0].order?.name && (
  31     <span className={styles.name}>
  32       | <b>{order[0].order.name}</b>
  33     </span>
  34   )}
  35
  36   {order.map( item => (
  37     <section className={styles.item} key={item.id}>
  38       <span>Qtd: {item.amount} - <b>{item.product.name}</b></span>
  39       <span className={styles.description}>
  40         | {item.product.description}</span>
  41       </section>
  42     ))
  43
  44   <button className={styles.buttonOrder} onClick={handleFinishOrder}>
  45     | Concluir pedido
  46   </button>
  47
  48   </article>
  49
  50   </section>
  51
  52 </dialog>
  53 )
  54 }
```

Browser Preview:

- Title:** Sujeito Pi
- URL:** localhost:3001
- Content:** A modal window titled "Detalhes do pedido" (Order Details) is displayed. It shows the table number (Mesa 35), the quantity and description for two "Burger Triplo Dev" items, and one "Dev Burger Salada" item. A green "Concluir pedido" (Finish Order) button is at the bottom.

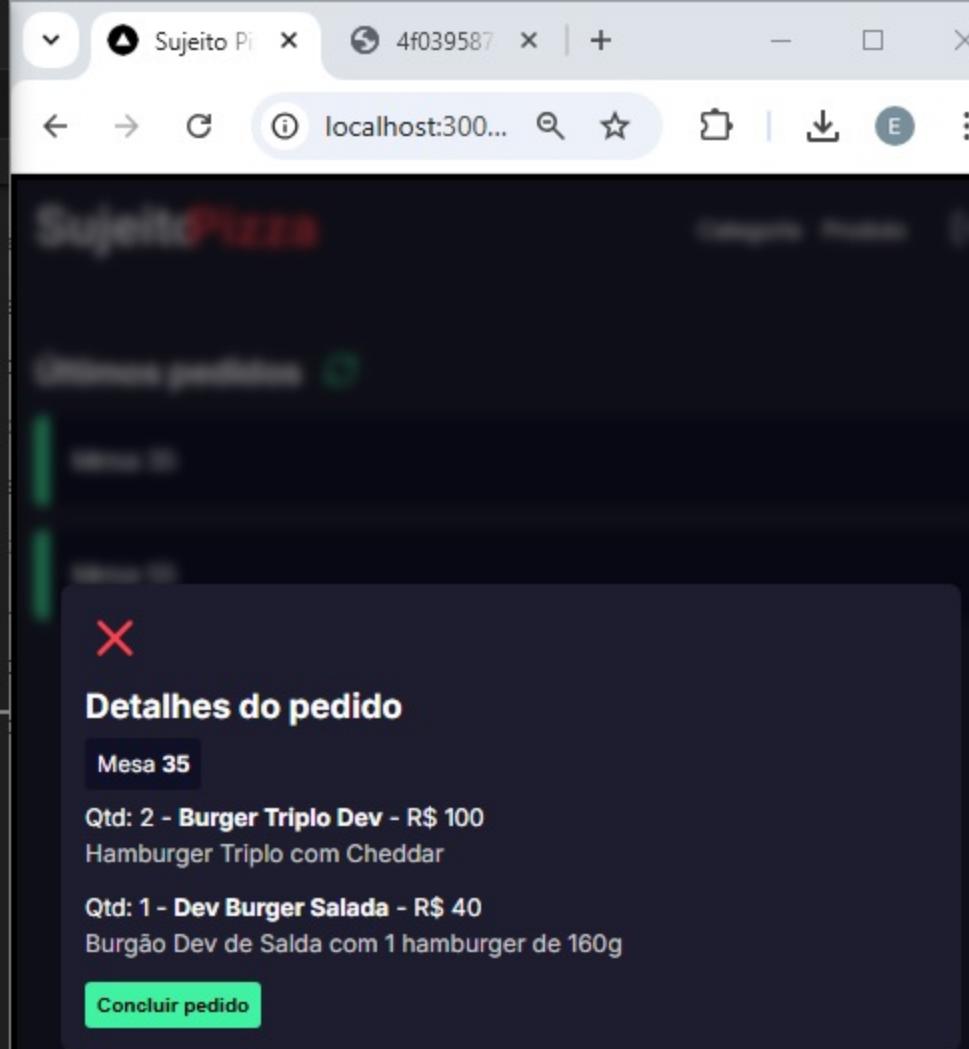
frontend

```
index.tsx ...\\orders index.tsx ...\\modal X
src > app > dashboard > components > modal > index.tsx > Modalorder > order.map() callback
26   <span className={styles.table}>
27     Mesa <b>{order[0].order.table}</b>
28   </span>
29
30   {order[0].order?.name && (
31     <span className={styles.name}>
32       <b>{order[0].order.name}</b>
33     </span>
34   )}
35
36   {order.map( item => (
37     <section className={styles.item} key={item.id}>
38       <span>
39         Qtd: {item.amount} - <b>{item.product.name}</b>
40       </span>
41       <span className={styles.description}>
42         {item.product.description}</span>
43     </section>
44   )));
45
46   <button className={styles.buttonOrder} onClick={handleFinishOrder}>
47     Concluir pedido
48   </button>
49
50   </article>
51
52   </section>
53
54   </dialog>
55
56 }
```



frontend

```
index.tsx ...\\orders index.tsx ...\\modal X
src > app > dashboard > components > modal > index.tsx > Modalorder > order.map() callback
26   <span className={styles.table}>
27     Mesa <b>{order[0].order.table}</b>
28   </span>
29
30   {order[0].order?.name && (
31     <span className={styles.name}>
32       <b>{order[0].order.name}</b>
33     </span>
34   )}
35
36   {order.map( item => (
37     <section className={styles.item} key={item.id}>
38       <span>
39         Qtd: {item.amount} - <b>{item.product.name}</b> - R$ {parseFloat(item.product.
40           price) * item.amount}
41       </span>
42       <span className={styles.description}>
43         {item.product.description}
44       </span>
45     ))
46
47     <button className={styles.buttonOrder} onClick={handleFinishOrder}>
48       Concluir pedido
49     </button>
50   </article>
51
52 </section>
53
54 </dialog>
55
56 }
```



frontend

```
index.tsx ...\\orders index.tsx ...\\modal X
src > app > dashboard > components > modal > index.tsx > Modalorder > order.map() callback
26   <span className={styles.table}>
27     | Mesa <b>{order[0].order.table}</b>
28   </span>
29
30   {order[0].order?.name && (
31     <span className={styles.name}>
32       | <b>{order[0].order.name}</b>
33     </span>
34   )}
35
36   {order.map( item => (
37     <section className={styles.item} key={item.id}>
38       <span>
39         | Qtd: {item.amount} - <b>{item.product.name}</b> - R$ {parseFloat(item.product.
40           price) * item.amount}
41       </span>
42       <span className={styles.description}>
43         | {item.product.description}
44       </span>
45     </section>
46   ))
47
48   <button className={styles.buttonOrder} onClick={handleFinishOrder}>
49     | Concluir pedido
50   </button>
51
52   </article>
53
54 </dialog>
55 )
56 }
```

Ln 39, Col 62 (52 selected) Spaces: 2 UTF-8 CRLF ⓘ TypeScript JSX ✅ Prettier

Sujeito Pizza 4f039587

localhost:300...

SujeitoPizza

Categoria Produto [→]

Últimos pedidos ⌂

- Mesa 35
- Mesa 55

The screenshot shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

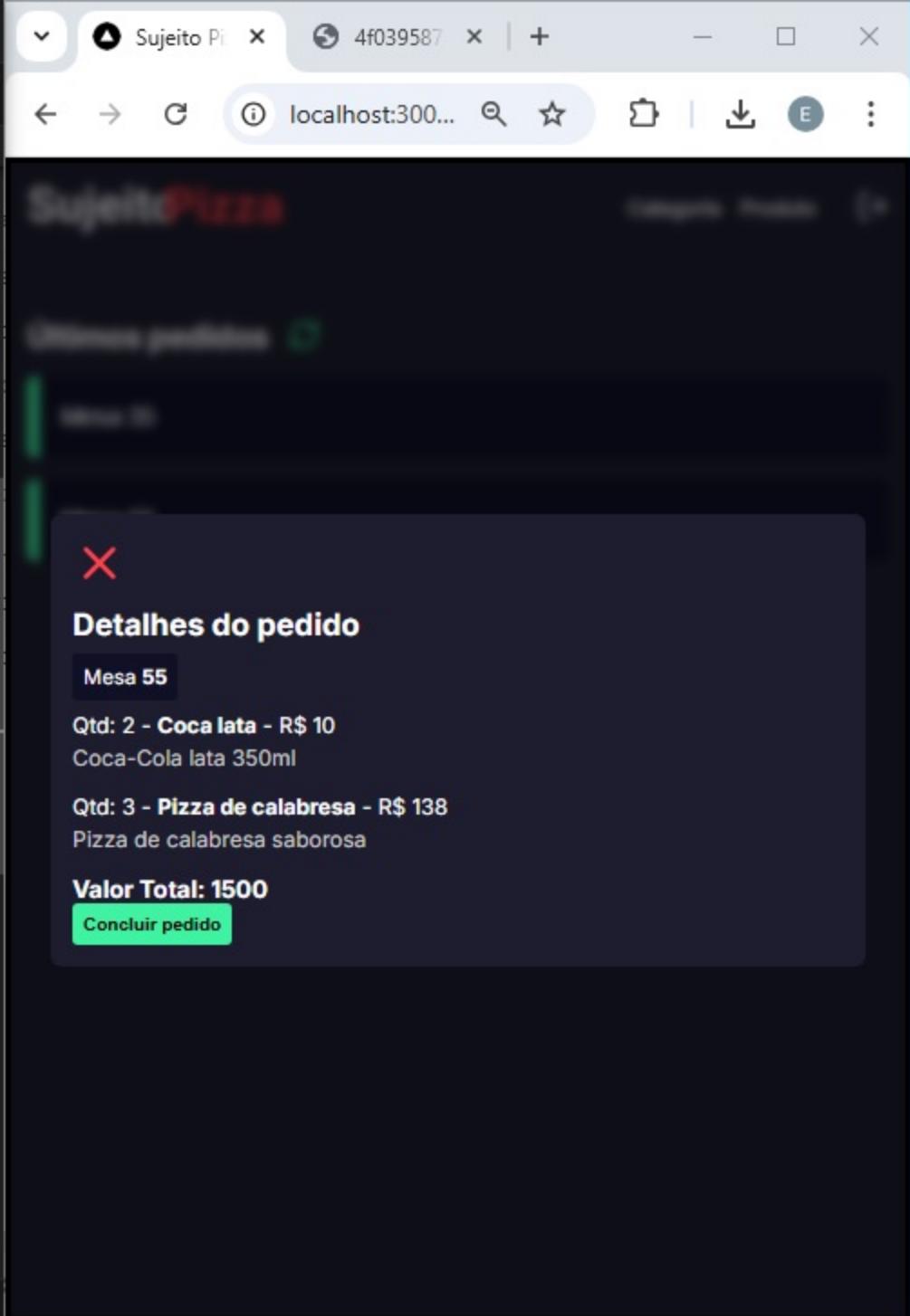
- Title Bar:** Shows the file path: `src > app > dashboard > components > modal > index.tsx` and the modal component name: `Modalorder`.
- Search Bar:** Contains the text `frontend`.
- Code Area:** Displays the `index.tsx` file content. The code is a modal component for managing orders. It includes logic to calculate the total price of items based on quantity and price, and a button to finish the order.
- Status Bar:** Shows the current line (Ln 39), column (Col 62), number of selected lines (52), and file encoding (UTF-8). It also indicates the file is a TypeScript/JSX file and is prettified.

Browser Preview:

- Title Bar:** Shows the title `Sujeito Pi` and the URL `localhost:300...`.
- Content:** Displays a modal dialog titled "Detalhes do pedido".
 - Mesa:** Mesa 55
 - Items:**
 - Qtd: 2 - Coca lata - R\$ 10
Coca-Cola lata 350ml
 - Qtd: 3 - Pizza de calabresa - R\$ 138
Pizza de calabresa saborosa
 - Buttons:** A green "Concluir pedido" button.

frontend

```
index.tsx ...\\orders index.tsx ...\\modal X
src > app > dashboard > components > modal > index.tsx > Modalorder
26   <span className={styles.table}>
27     Mesa <b>{order[0].order.table}</b>
28   </span>
29
30   {order[0].order?.name && (
31     <span className={styles.name}>
32       <b>{order[0].order.name}</b>
33     </span>
34   )}
35
36   {order.map( item => (
37     <section className={styles.item} key={item.id}>
38       <span>
39         Qtd: {item.amount} - <b>{item.product.name}</b> - R$ {parseFloat(item.product.
40           price) * item.amount}
41       </span>
42       <span className={styles.description}>
43         {item.product.description}
44       </span>
45     </section>
46   ))}
47
48   <h3 className={styles.total}>Valor Total: 1500 </h3>
49
50   <button className={styles.buttonOrder} onClick={handleFinishOrder}>
51     Concluir pedido
52   </button>
53
54 </article>
55
56 </dialog>
57
```



frontend

EXPLORER

FRONTEND

- > .next
- > node_modules
- > public
- src
 - app
 - dashboard
 - > category
 - > components
 - button
 - > header
 - > modal
 - > index.tsx
 - styles.module.scss
 - > orders
 - > product
 - layout.tsx
 - page.tsx
 - > fonts
 - > signup
 - favicon.ico
 - globals.scss
 - layout.tsx
 - page.module.scss
 - page.tsx
 - > lib
 - > providers
- > OUTLINE
- > TIMELINE

index.tsx ...\\orders index.tsx ...\\modal styles.module.scss X

src > app > dashboard > components > modal > styles.module.scss > ...

```
51
52     .description{
53         color: #CCC;
54         margin-top: 4px;
55     }
56
57     .buttonOrder{
58         background-color: var(--green-900);
59         padding: 8px;
60         border-radius: 4px;
61         border: 0;
62         font-weight: bold;
63     }
64
65     .name {
66         margin-left: 16px;
67     }
68
69     .total {
70         margin: 8px 0 16px 0;
71     }
```

Ln 69, Col 1 (42 selected) Spaces: 2 UTF-8 CRLF { SCSS ✓ Prettier

Sujeito Pi x 4f039587 x +

localhost:300...

SuperaPizz

Novo pedido

Mesa 55

X

Detalhes do pedido

Mesa 55

Qtd: 2 - Coca lata - R\$ 10
Coca-Cola lata 350ml

Qtd: 3 - Pizza de calabresa - R\$ 138
Pizza de calabresa saborosa

Valor Total: 1500

Concluir pedido

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure. The **FRONTEND** folder is expanded, showing **.next**, **node_modules**, **public**, and **src**. The **src** folder contains **app**, **dashboard**, **category**, **components**, **button**, **header**, and **modal**. The **index.tsx** file under **modal** is currently selected.
- Search:** Shows the search results for **frontend**.
- File:** Shows the file tree with files like **index.tsx**, **styles.module.scss**, **orders**, **product**, **layout.tsx**, **page.tsx**, **fonts**, **signup**, **favicon.ico**, **globals.scss**, **layout.tsx**, **page.module.scss**, **page.tsx**, **lib**, **providers**, **OUTLINE**, and **TIMELINE**.
- Editor:** Displays the content of **index.tsx** (Modalorder component). The code uses TypeScript and JSX to render a modal dialog with a table of items and a total value.
- Bottom Status Bar:** Shows the current position as **Ln 46, Col 60 (7 selected)**, **Spaces: 2**, **UTF-8**, **CRLF**, **TypeScript JSX**, and **Prettier**.

Browser Preview:

- Title:** Sujeto Pi
- URL:** localhost:3007
- Content:** A modal dialog titled "Detalhes do pedido" (Details of the order) is displayed. It shows the table from the code editor, indicating 2 Coca-Cola lata (350ml) at R\$ 10 each, totaling R\$ 1500. A button labeled "Concluir pedido" (Finish order) is visible.

The image shows a developer's workspace with two main windows: a code editor and a web browser.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
 - src
 - app
 - dashboard
 - fonts
 - signup
 - favicon.ico
 - globals.scss
 - layout.tsx
 - page.module.scss
 - page.tsx
 - lib
 - cookieClient.ts
 - cookieServer.ts
 - helper.ts
 - order.type.ts
 - providers
 - services
 - middleware.ts
 - .gitignore
 - next-env.d.ts
 - next.config.ts
 - package-lock.json
 - package.json
- Search:** Shows search results for "frontend".
- Outline:** Shows the outline of the current file.
- Timeline:** Shows the timeline of the current file.

Browser:

- Title:** Sujeto Pi
- URL:** localhost:300...
- Content:** A modal window titled "Detalhes do pedido" for "Mesa 55". It lists items:
 - Qtd: 2 - Coca lata - R\$ 10
Coca-Cola lata 350ml
 - Qtd: 3 - Pizza de calabresa - R\$ 138
Pizza de calabresa saborosa**Value Total:** R\$ 1500
- Buttons:** Concluir pedido

The image shows a developer's environment with two main windows: a code editor in the foreground and a browser preview in the background.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** "TS helper.ts" is open, containing the following TypeScript code:

```
1  export function calculateTotalOrder(orders:[]){
2    ...
3  }
```
- Status Bar:** Shows "Ln 4, Col 2 (51 selected)" and other file-related information.

Browser Preview:

- Title Bar:** "Sujeito Pi" and "4f039587".
- URL:** "localhost:300...".
- Content:** A dark-themed UI component titled "Detalhes do pedido" (Order Details). It displays:
 - Mesa 55
 - Qtd: 2 - Coca lata - R\$ 10
Coca-Cola lata 350ml
 - Qtd: 3 - Pizza de calabresa - R\$ 138
Pizza de calabresa saborosa
 - Valor Total: R\$ 1500**
 - A green "Concluir pedido" button.

frontend

EXPLORER

FRONTEND

- > .next
- > node_modules
- > public
- src
 - app
 - > dashboard
 - > fonts
 - > signup
 - favicon.ico
 - globals.scss
 - layout.tsx
 - page.module.scss
 - page.tsx
 - lib
 - TS cookieClient.ts
 - TS cookieServer.ts
 - TS helper.ts 1
 - TS order.type.ts
 - providers
 - order.tsx
 - > services
 - TS middleware.ts
 - .gitignore
 - TS next-env.d.ts
 - TS next.config.ts
 - { package-lock.json
- > OUTLINE
- > TIMELINE

index.tsx ...\\orders index.tsx ...\\modal TS helper.ts 1 order.tsx ...

```
src > providers > order.tsx > OrderItemProps
  1 "use client"
  2 import { createContext, ReactNode, useState } from 'react'
  3 import { api } from '@/services/api';
  4 import { getCookieClient } from '@/lib/cookieClient';
  5 import { toast } from 'sonner';
  6 import { useRouter } from 'next/navigation';
  7
  8 export interface OrderItemProps{
  9   id: string;
 10   amount: number;
 11   created_at: string;
 12   order_id: string;
 13   product_id: string;
 14   product:{
 15     id: string;
 16     name: string;
 17     price: string;
 18     description: string;
 19     banner: string;
 20     category_id: string;
 21   };
 22   order:{
 23     id: string;
 24     table: number;
 25     name: string | null;
 26     draft: boolean;
 27     status: boolean;
 28   }
 29 }
 30
 31 type OrderContextData = {
 32   isOpen: boolean;
 33 }
```

Ln 8. Col 1 (32 selected) Spaces: 2 CRLF ↴ TypeScript JSX ✅ Prettier

Sujeito Pizza

Categoria Produto [→]

Últimos pedidos ↗

Mesa 35

Mesa 55

The image shows a developer's environment with two main windows: a code editor in the foreground and a browser preview in the background.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** index.tsx (selected), modal, helper.ts, order.tsx.
- Content:** The code is for a modal component displaying order details. It includes logic to handle the order table and calculate the total value based on quantity and price.

```
index.tsx ...\\orders
src > app > dashboard > components > modal > index.tsx > Modalorder > order.map() callback
26   <span className={styles.table}>
27     | Mesa <b>{order[0].order.table}</b>
28   </span>
29
30   {order[0].order?.name && (
31     <span className={styles.name}>
32       | <b>{order[0].order.name}</b>
33     </span>
34   )}
35
36   {order.map( item => (
37     <section className={styles.item} key={item.id}>
38       <span>
39         | Qtd: {item.amount} - <b>{item.product.name}</b> - R$ {parseFloat(item.product.price) * item.amount}
40       </span>
41       <span className={styles.description}>
42         | {item.product.description}</span>
43     </section>
44   ))}
45
46   <h3 className={styles.total}>Valor Total: R$ 1500 </h3>
47
48   <button className={styles.buttonOrder} onClick={handleFinishOrder}>
49     | Concluir pedido
50   </button>
51
52   </article>
53
54 </section>
55
56 </dialog>
```

Browser Preview:

- Title:** SuaPedra
- Content:** A modal window titled "Detalhes do pedido" (Order Details) is displayed. It shows the table number (Mesa 35), the items ordered (Burger Triplo Dev and Dev Burger Salada), their quantities, and the total value (R\$ 1500). A green button at the bottom right says "Concluir pedido" (Finish Order).

frontend

index.tsx ...\\orders index.tsx ...\\modal TS helper.ts order.tsx

src > lib > TS helper.ts > ...

```
1 import { OrderItemProps } from "@/providers/order";
2
3 export function calculateTotalOrder(orders: OrderItemProps[]){
4   return orders.reduce((total, item) => {
5     const itemTotal = parseFloat(item.product.price) * item.
6       amount
7     return total + itemTotal
8   }, 0)
}
```

EXPLORER

FRONTEND

- src
 - app
 - dashboard
 - components
 - orders
 - product
 - layout.tsx
 - page.tsx
 - fonts
 - signup
 - favicon.ico
 - globals.scss
 - layout.tsx
 - page.module.scss
 - page.tsx
- lib
 - cookieClient.ts
 - cookieServer.ts
 - TS helper.ts
 - order.type.ts
- providers
 - order.tsx
- services
- middleware.ts
- .gitignore
- next-env.d.ts

- OUTLINE
- TIMELINE

Ln 1, Col 1 (282 selected) Spaces: 4 UTF-8 CRLF TypeScript Prettier

Sujeito Pi 4f039587

localhost:300...

Detalhes do pedido

Mesa 35

Qtd: 2 - **Burger Triplo Dev** - R\$ 100
Hambúrguer Triplo com Cheddar

Qtd: 1 - **Dev Burger Salada** - R\$ 40
Burgão Dev de Salada com 1 hambúrguer de 160g

Valor Total: R\$ 1500

Concluir pedido

frontend

index.tsx ...\\orders index.tsx ...\\modal X TS helper.ts order.tsx ...

src > app > dashboard > components > modal > index.tsx > ...

```
1 "use client"
2
3 import styles from './styles.module.scss'
4 import { Import, X } from 'lucide-react'
5 import { use } from 'react'
6 import { OrderContext } from '@/providers/order'
7 import { calculateTotalOrder } from '@/lib/helper'
8
9 export function Modalorder(){
10   const { onRequestClose, order, finishOrder } = use(OrderContext);
11
12   async function handleFinishOrder() {
13     await finishOrder(order[0].order.id)
14   }
15
16   return(
17     <dialog className={styles.dialogContainer}>
18
19       <section className={styles.dialogContent}>
20         <button className={styles.dialogBack} onClick={onRequestClose}>
21           <X size={40} color="#FF3f4b" />
22         </button>
23
24         <article className={styles.container}>
25           <h2>Detalhes do pedido</h2>
26
27           <span className={styles.table}>
28             Mesa <b>{order[0].order.table}</b>
29           </span>
30
31           {order[0].order?.name && (
32             <div>
33               <h3>Nome:</h3>
34               <p>{order[0].order.name}</p>
35             </div>
36           )}
37         </article>
38       </section>
39     </dialog>
40   )
41 }
```

Ln 7, Col 1 (50 selected) Spaces: 2 CRLF ⓘ TypeScript JSX ✓ Prettier

Sujeito Pizza

Categoria Produto [→]

Últimos pedidos ↻

Mesa 35

Mesa 55

The screenshot shows the Visual Studio Code interface with the following details:

- Explorer View:** Shows the project structure under "FRONTEND".
- Code Editor:** Displays the file `index.tsx` from the `src/app/dashboard/components/modal` directory. The code is a modal component for displaying orders.
- Status Bar:** Shows the current line (Ln 47), column (Col 60), and selected text count (4 selected).

```
index.tsx ...\\orders index.tsx ...\\modal X TS helper.ts order.tsx ...  
src > app > dashboard > components > modal > index.tsx > Modalorder  
34     </span>  
35     )}  
36  
37     {order.map( item => (  
38         <section className={styles.item} key={item.id}>  
39             <span>  
40                 Qtd: {item.amount} - <b>{item.product.name}</b> - R$  
41                 {parseFloat(item.product.price) * item.amount}  
42             </span>  
43             <span className={styles.description}>  
44                 {item.product.description}</span>  
45         </section>  
46     ))}  
47  
48     <h3 className={styles.total}>Valor Total: R$ 1500 </h3>  
49  
50     <button className={styles.buttonOrder} onClick={handleFinishOrder}>  
51         Concluir pedido  
52     </button>  
53  
54     </article>  
55  
56     </section>  
57  
58     </dialog>  
59 }
```

The screenshot shows a browser window displaying the `Sujeito Pizza` website at `localhost:3001`. The page features a header with the logo and navigation links for "Categoria" and "Produto". Below the header, there is a section titled "Últimos pedidos" (Last orders) with two items listed:

- Mesa 35
- Mesa 55

The image shows a developer's workspace with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** index.tsx (selected), styles.module.scss, orders, product, layout.tsx, page.tsx, fonts, signup, favicon.ico, globals.scss, layout.tsx, page.module.scss, page.tsx, lib, cookieClient.ts.
- Search:** frontend
- Editor:** Content of index.tsx (Modalorder component). The code uses TypeScript and JSX. It maps through an order array, rendering sections for each item with details like quantity, product name, and total price. It also includes a button to handle the finish order.
- Status Bar:** Ls 47, Col 84 (28 selected) Spaces: 2 UTF-8 CRLF { TypeScript JSX ✅ Prettier

Browser Preview:

- Title:** Sujeito Pi x 4f039587
- URL:** localhost:300...
- Content:** A modal window titled "Detalhes do pedido" (Details of the order) for "Mesa 35". It lists two items:
 - Qtd: 2 - Burger Triplo Dev - R\$ 100
Hamburger Triplo com Cheddar
 - Qtd: 1 - Dev Burger Salada - R\$ 40
Burgão Dev de Salada com 1 hambúrguer de 160gA large green button at the bottom right says "Concluir pedido" (Finish order).

frontend

index.tsx ... \orders index.tsx ... \modal TS helpers.ts order.tsx

```
src > app > dashboard > components > modal > index.tsx > Modalorder
34         </span>
35     )
36
37     {order.map( item => (
38         <section className={styles.item} key={item.id}>
39             <span>
40                 Qtd: {item.amount} - <b>{item.product.name}</b> - R$ {parseFloat(item.product.price) * item.amount}
41             </span>
42             <span className={styles.description}>
43                 {item.product.description}</span>
44             </section>
45     ))}
46
47     <h3 className={styles.total}>Valor Total: R$ {calculateTotalOrder(order)} </h3>
48
49     <button className={styles.buttonOrder} onClick={handleFinishOrder}>
50         Concluir pedido
51     </button>
52
53     </article>
54
55     </section>
56
57     </dialog>
58
59 )
```

EXPLORER

FRONTEND

- > .next
- > node_modules
- > public
- src
 - app
 - dashboard
 - > category
 - > components
 - > button
 - > header
 - > modal
 - index.tsx
 - styles.module.scss

Sujeito Pi x 4f039587 +

localhost:300... 🔍 ⚡ ⏪ ⏴ E ⏵

SujeitoPizza

Categoria Produto [→]

Últimos pedidos ⌂

- Mesa 35
- Mesa 55

The image shows a developer's environment with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure under "FRONTEND".
- File:** index.tsx (selected), helper.ts, orders.tsx.
- Content:** The code is for a modal component displaying order details. It includes mapping over an order array, rendering sections for each item with descriptions and prices, and a total value. A button is provided to finish the order.

```
index.tsx ...\\orders
src > app > dashboard > components > modal > index.tsx > Modalorder
34     </span>
35   )
36
37   {order.map( item => (
38     <section className={styles.item} key={item.id}>
39       <span>
40         Qtd: {item.amount} - <b>{item.product.name}</b> - R$ {parseFloat(item.product.price) * item.amount}
41       </span>
42       <span className={styles.description}>
43         {item.product.description}</span>
44     </section>
45   ))}

<h3 className={styles.total}>Valor Total: R$ {calculateTotalOrder(order)} </h3>

<button className={styles.buttonOrder} onClick={handleFinishOrder}>
  Concluir pedido
</button>
```

Browser Preview:

- Title:** Sujeto Pi
- URL:** localhost:300...
- Content:** A modal dialog titled "Detalhes do pedido" (Order Details) is displayed. It shows the table number (Mesa 55), the items ordered (2 Coca lata - R\$ 10, Coca-Cola lata 350ml), and the total value (Valor Total: R\$ 148). A green "Concluir pedido" (Finish Order) button is at the bottom.

Bottom status bar: Ln 47, Col 84 (28 selected) Spaces: 2 CRLF TypeScript JSX Prettier

The image shows a developer's workspace with two main windows: a code editor and a browser preview.

Code Editor (VS Code):

- Explorer:** Shows the project structure. The **FRONTEND** folder contains **.next**, **node_modules**, **public**, and **src**. The **src** folder contains **app**, **dashboard**, **category**, **components**, **button**, **header**, **modal**, **index.tsx** (selected), **styles.module.scss**, **orders**, **product**, **layout.tsx**, **page.tsx**, **fonts**, **signup**, **favicon.ico**, **globals.scss**, **layout.tsx**, **page.module.scss**, **page.tsx**, **lib**, and **cookieClient.ts**.
- Search:** Shows the search results for **frontend**, listing **index.tsx**, **helper.ts**, **order.tsx**, and **...**.
- Editor:** The **index.tsx** file is open, showing TypeScript code for rendering a modal. The code includes mapping over an **order** array, rendering sections for each item with descriptions and prices, and calculating a total value. It also includes a button to handle the finish order.

Browser Preview:

The browser window displays a dark-themed interface for a food delivery application. A modal dialog is open, titled "Detalhes do pedido" (Order Details). The dialog shows the following information:

- Mesa 55**
- Qtd: 2 - Coca lata - R\$ 10**
Coca-Cola lata 350ml
- Qtd: 3 - Pizza de calabresa - R\$ 138**
Pizza de calabresa saborosa
- Valor Total: R\$ 148**
- Concluir pedido** button

The browser address bar shows `localhost:300...`.

frontend

index.tsx ... \orders index.tsx ... \modal X TS helper.ts order.tsx ...

src > app > dashboard > components > modal > index.tsx > Modalorder

```
34         </span>
35     )
36
37     {order.map( item => (
38         <section className={styles.item} key={item.id}>
39             <span>
40                 Qtd: {item.amount} - <b>{item.product.name}</b> - R$ {parseFloat(item.product.price) * item.amount}
41             </span>
42             <span className={styles.description}>
43                 {item.product.description}</span>
44             </section>
45     ))}

46
47     <h3 className={styles.total}>Valor Total: R$ {calculateTotalOrder(order)} </h3>
48
49     <button className={styles.buttonOrder} onClick={handleFinishOrder}>
50         Concluir pedido
51     </button>
52
53 </article>
54
55 </section>
56
57 </dialog>
58
59 )
```

EXPLORER

FRONTEND

- > .next
- > node_modules
- > public
- src
 - app
 - dashboard
 - > category
 - > components
 - > button
 - > header
 - modal
 - > index.tsx
 - > styles.module.scss

Sujeito Pizza

Categoria Produto [→]

Últimos pedidos

Mesa 35

Pedido finalizado com sucesso!