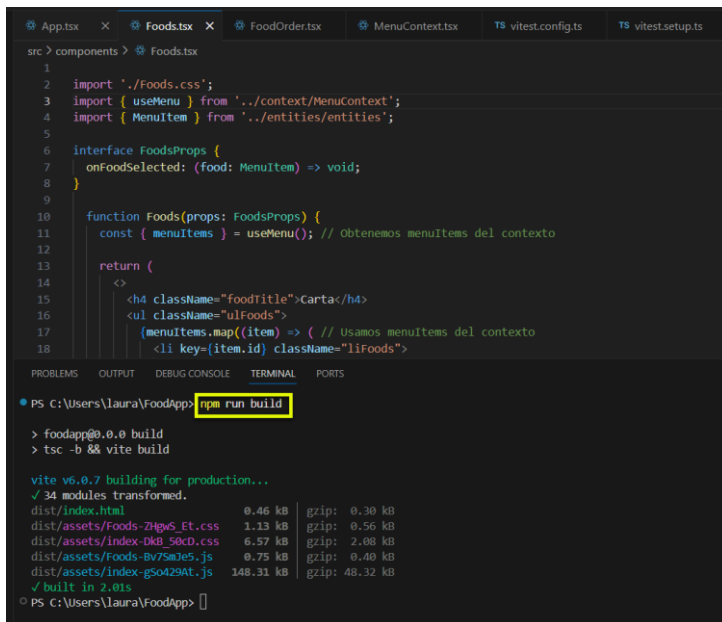


Capturas funcionamiento Despligue:



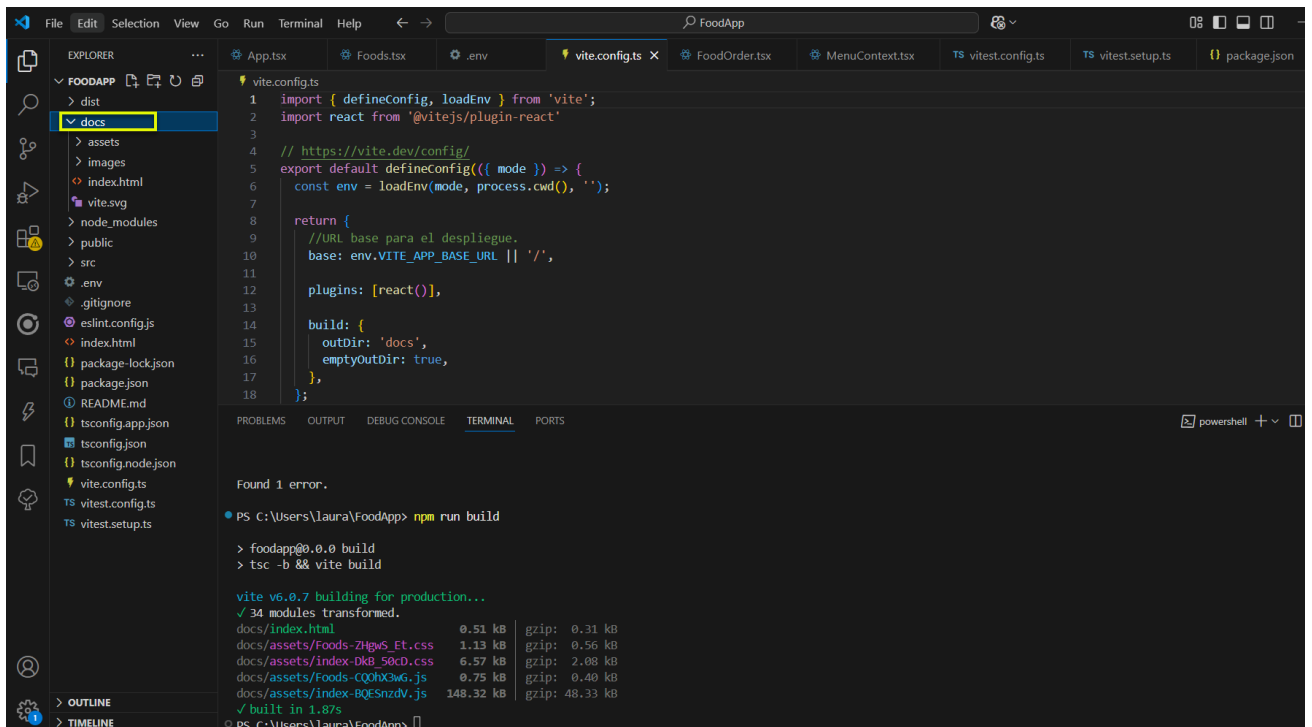
```
src > components > Food.tsx
1
2 import './Foods.css';
3 import { useMenu } from '../context/MenuContext';
4 import { MenuItem } from '../entities/entities';
5
6 interface FoodsProps {
7   onFoodSelected: (food: MenuItem) => void;
8 }
9
10 function Foods(props: FoodsProps) {
11   const { menuItems } = useMenu(); // Obtenemos menuItems del contexto
12
13   return (
14     <>
15       <h4 className="foodTitle">Carta</h4>
16       <ul className="ulFoods">
17         {menuItems.map((item) => ( // Usamos menuItems del contexto
18           <li key={item.id} className="liFoods">
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\laura\FoodApp> npm run build
> foodapp@0.0.0 build
> tsc -b && vite build

vite v6.0.7 building for production...
✓ 34 modules transformed.
dist/index.html      0.46 kB | gzip: 0.30 kB
dist/assets/Foods-ZHgw5_Et.css 1.13 kB | gzip: 0.56 kB
dist/assets/index-DkB_50cD.css 6.57 kB | gzip: 2.08 kB
dist/assets/Foods-Bv7SmJe5.js 0.75 kB | gzip: 0.40 kB
dist/assets/index-g5o429at.js 148.31 kB | gzip: 48.32 kB
✓ built in 2.01s
PS C:\Users\laura\FoodApp>
```

Después de modificar el archivo “vite.config.ts” ejecutamos nuevamente npm run build, crea una carpeta llamada “docs” en la raíz del proyecto. Esta carpeta contendrá todos los archivos estáticos listos para ser desplegados (index.html, CSS, JavaScript, imágenes, etc.)



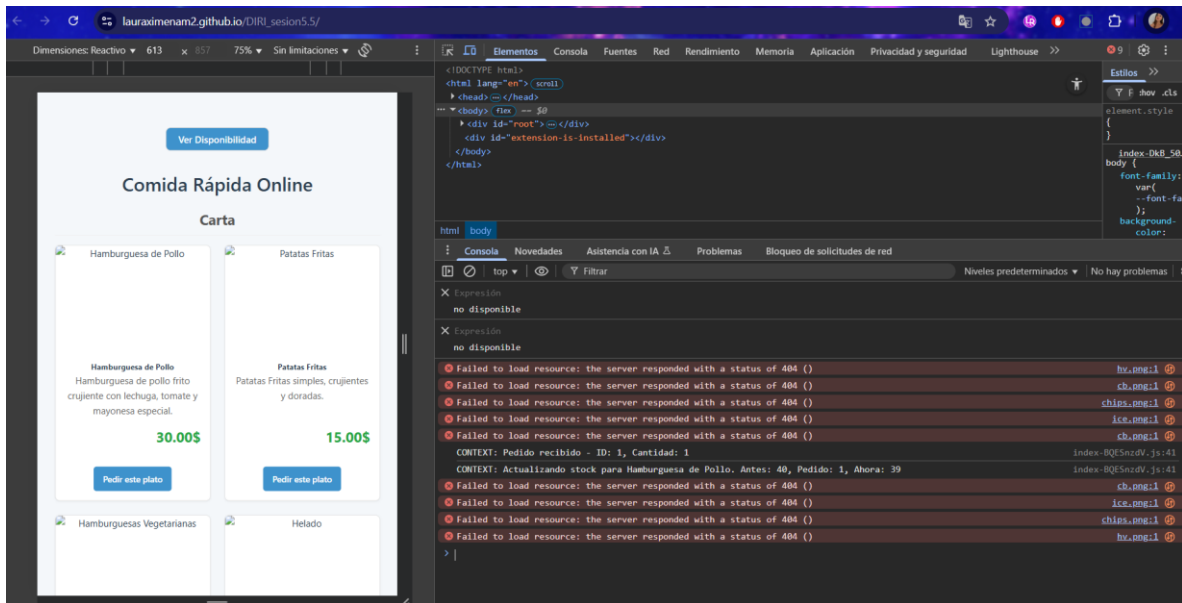
```
File Edit Selection View Go Run Terminal Help
FoodApp
vite.config.ts
1 import { defineConfig, loadEnv } from 'vite';
2 import react from '@vitejs/plugin-react'
3
4 // https://vite.dev/config/
5 export default defineConfig(({ mode }) => {
6   const env = loadEnv(mode, process.cwd(), '');
7
8   return {
9     //URL base para el despliegue.
10    base: env.VITE_APP_BASE_URL || '/',
11
12    plugins: [react()],
13
14    build: {
15      outDir: 'docs',
16      emptyOutDir: true,
17    },
18  };
19
20
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Found 1 error.
PS C:\Users\laura\FoodApp> npm run build
> foodapp@0.0.0 build
> tsc -b && vite build

vite v6.0.7 building for production...
✓ 34 modules transformed.
docs/index.html      0.51 kB | gzip: 0.31 kB
docs/assets/Foods-ZHgw5_Et.css 1.13 kB | gzip: 0.56 kB
docs/assets/index-DkB_50cD.css 6.57 kB | gzip: 2.08 kB
docs/assets/Foods-CQ0hX3wG.js 0.75 kB | gzip: 0.40 kB
docs/assets/index-BQESnzdv.js 148.32 kB | gzip: 48.33 kB
✓ built in 1.87s
PS C:\Users\laura\FoodApp>
```

Subimos los cambios a GitHub

Se genera la URL pública de la app de comida:
https://lauraximenam2.github.io/DIRI_sesion5.5/

Al ejecutarla obtenemos un problema, y es que las imágenes no cargan correctamente.



Para solucionar esto, realizamos lo siguiente:

- Modificamos la forma de cómo se establece el src de las imágenes en Foods.tsx y FoodOrder.tsx:
- Revisamos la configuración de “base” en nuestro archivo “vite.config.ts”, que tenga EXACTAMENTE el mismo nombre del repositorio.
- En .env revisamos también la línea: VITE_APP_BASE_URL=/DIRI_sesion5.5/

```
App.tsx  Foods.tsx  FoodOrder.tsx X  .env  vite.config.ts  <>

src > components > FoodOrder.tsx
12  function FoodOrder(props: FoodOrderProps) {
57
58      useEffect(() => {
59          setConfirmationMessage('');
60          setOrderQuantity(1);
61          setIsOrderSent(false);
62      }, [props.food]);
63
64      return (
65          <div className="foodOrderContainer">
66              <h3>Detalles del Pedido</h3>
67              <div className="orderDetails">
68                  <img
69                      className="orderFoodImg"
70                      src={` ${import.meta.env.BASE_URL}images/${item.image}`}
71                      alt={props.food.name}
72                  />
73              </div>
74          </div>
75      );
76  }
77  export default FoodOrder;
```

```
File  Edit  Selection  View  Go  Run  Terminal  Help  FoodApp
EXPLORER
FOODAPP
  docs
  images
    chips.png
    hv.png
    ice.png
  index.html
  vite.svg
  node_modules
  public
    images
      cb.png
      chips.png
      hv.png
      ice.png
      vite.svg
  src
    _tests_
    assets
    components
      FoodOrder.tsx

App.tsx  Foods.tsx  FoodOrder.tsx X  .env  vite.config.ts  index.html

src > components > FoodOrder.tsx
12  function FoodOrder(props: FoodOrderProps) {
64      return (
65          <div className="foodOrderContainer">
66              <div className="orderDetails">
67                  <img
68                      className="orderFoodImg"
69                      src={` ${import.meta.env.BASE_URL}images/${props.food.image}`}
70                      alt={props.food.name}
71                  />
72              </div>
73              <div className="orderInfo">
74                  <h4>{props.food.name}</h4>
75                  <p className="orderDesc">{props.food.desc}</p>
76                  <p className="orderBasePrice">Precio unitario: {props.food.price.toFixed(2)}$</p>
77                  <div className="quantityControl">
78                      <label htmlFor={`quantity-${props.food.id}`}>Cantidad:</label>
79                      <input
80                          type="number"
81                          id={`quantity-${props.food.id}`}
82                          name="quantity"
83                          min="0"
84                          value={orderQuantity}
85                          onChange={handleQuantityChange}
86                      />
87                  </div>
73              </div>
74          </div>
75      );
76  }
77  export default FoodOrder;
```

Construimos nuevamente con `npm run build` y ahora cargan correctamente las imágenes:

