

DOCUMENTACION ECOMMERCE

DEPENDENCIAS

Bootstrap
Material UI
React Router Dom

LOGICA

Creamos el componente Item que es la card que se va a renderizar por cada uno de los objetos que traemos de data.js

```

return (
  <Link to={`/detalle/${info.id}`}>
    <Card sx={{ maxWidth: 345 }}>
      <CardActionArea>
        <CardMedia
          component="img"
          height="340"
          image={info.img}
          alt="Cinta de Correr"
          stock={info.stock}
        />
        <CardContent>
          <Typography gutterBottom variant="h6" component="div">
            {info.title}
          </Typography>
          <Typography variant="body2" color="text.secondary">
            {info.desc}
          </Typography>
          <div className="Container">
            <Typography variant="body1" color="text.primary">
              {info.price}
            </Typography>
            <ItemCount stock={info.stock}/>
          </div>
        </CardContent>
      </CardActionArea>
    </Card>
  </Link>
);

```

En ItemList mapeamos cada producto y generamos un Item.

```
const ItemList = ({data = []}) => {  
  return (  
    | | data.map(cinta => <div className="ItemList"><Item key={cinta.id}  
    | );  
  );  
};  
  
export default ItemList;
```

Y el componente ItemListContainer que es el contenedor donde se maneja toda la lógica.
Por medio de una promesa nos devuelve los productos de data.js
Utilizamos useState y useEffect para simular un ciclo de vida.

```

function App() {
  return (
    <React.Fragment>
      <Router>
        <div className="App">
          <NavBar />
        </div>
        <div className="Greeting">
          <Greeting
            nombre="Carla"
            img="../assets/images/usuario.png"/>
        </div>
        <Routes>
          <Route path="/" element={<div className="UserSection"><ItemL
            <Route path="/categoria/:categoriaId" element={<div classN
            <Route path="/detalle/:detalleId" element={<ItemDetailCont
          </Route>
        </Routes>
      </Router>
    </React.Fragment>
  )
}

```

En App.js se define la estructura de la aplicación.
 Envolveremos con BrowserRouter la app, y definiremos las
 rutas con el componente Routes.
 Nav-Bar y Greeting quedan fuera de Routes porque
 queremos que se vean siempre.

```
function App() {  
  return (  
    <React.Fragment>  
      <Router>  
        <div className="App">  
          <NavBar />  
        </div>  
        <div className="Greeting">  
          <Greeting  
            nombre="Carla"  
            img="../../assets/images/usuario.png"/>  
        </div>  
        <Routes>  
          <Route path="/" element={<div className="UserSection"><ItemL  
            <Route path="/categoria/:categoriaId" element={<div classN  
            <Route path="/detalle/:detalleId" element={<ItemDetailCont  
          </Routes>  
        </Router>  
      </React.Fragment>  
    )  
  )  
}
```