



index.ts

```
export * from './NavBar';
export * from './Sidebar';
export * from './EntryList';
export * from './EntryCard';
export * from './NewEntry';
```

EntryCard.tsx

```
import React, { DragEvent, FC, useContext } from "react";
import { useRouter } from "next/router";

import { Card, CardActionArea, CardActions, CardContent,
  Typography, } from "@mui/material";

import { Entry } from "../../interfaces";
import { UIContext } from "../../context/ui";
import { dateFunctions } from "../../utils";

interface Props {
  entry: Entry;
}

export const EntryCard: FC<Props> = ({ entry }) => {

  const { startDragging, endDragging } = useContext( UIContext )
  const routes = useRouter();

  const onDragStart = ( event:DragEvent ) => {
    event.dataTransfer.setData('text', entry._id)

    startDragging();
  }

  const onDragEnd = () => {
    endDragging();
  }

  const onClick = () => {
    routes.push(`/entries/${entry._id}`)
  }

  return (
    <Card
      onClick={onClick}
      sx={{ marginBottom: 1 }}
    >
```

```

        draggable={true}
        onDragStart={ onDragStart }
        onDragEnd={ onDragEnd }
      >
        <CardActionArea>
          <CardContent>
            <Typography sx={{ whiteSpace: 'pre-line'}}>
              { entry.description }
            </Typography>
          </CardContent>

          <CardActions sx={{ display: 'flex',
            justifyContent: 'end', paddingRight: 2}}>

            <Typography variant="body2">
              {dateFunctions.
                getFormatDistanceToNow(entry.createdAt)}

            </Typography>
          </CardActions>
        </CardActionArea>
      </Card>
    );
  };
};

```

EntryList.module.css

```

.dragging {
  background-color: rgba(255, 255, 255, 0.2);
  border-radius: 10px;
  border: 1px dashed white;
}

```

EntryList.tsx

```

import { List, Paper } from '@mui/material'
import React, { DragEvent, FC, useContext, useMemo } from 'react'
import { EntryCard } from './'
import { EntryStatus } from '../interfaces';
import { EntriesContext } from '../context/entries';
import { UIContext } from '../context/ui';

import styles from './EntryList.module.css'

interface Props {
  status: EntryStatus;
}

export const EntryList:FC<Props> = ({ status }) => {
  const { entries, updateEntry } = useContext( EntriesContext )

  const { isDragging, endDragging } = useContext( UIContext )

  const entriesByStatus = useMemo(
    () => entries.filter((entry) => entry.status === status),
    [entries]
  );
};

```

```

const allowDrop = ( event:DragEvent<HTMLDivElement> ) => {
  event.preventDefault();
}

const onDropEntry = ( event:DragEvent<HTMLDivElement> ) => {
  const id = event.dataTransfer.getData('text')

  const entry = entries.find( e => e._id == id );
  entry.status = status;
  updateEntry( entry )
  endDragging();
}

return (
  <div
    onDrop={onDropEntry}
    onDragOver={allowDrop}
    className={isDragging ? styles.dragging : ""}>
    <Paper
      sx={{
        height: "calc(100vh - 180px)",
        overflow: "auto",
        backgroundColor: "transparent",
        padding: "3px 5px",
        paddingBottom: "2",
      }}>

      <List sx={{ opacity: isDragging ? 0.2 : 1, transition: "all 0.3s" }}>
        {entriesByStatus.map((entry) => (
          <EntryCard key={entry._id} entry={entry} />
        ))}
      </List>
    </Paper>
  </div>
);
}

```

NavBar.tsx

```

import { FC, useContext } from "react";

import NextLink from "next/link";

import { AppBar, IconButton, Link, Toolbar, Typography } from "@mui/material";
import MenuOutlinedIcon from "@mui/icons-material/MenuOutlined";
import { UIContext } from "../../context/ui";

export const NavBar: FC = () => {

  const { openSideMenu } = useContext( UIContext )
  return (
    <AppBar position="sticky">
      <Toolbar>
        <IconButton
          size="large"
          edge="start"
          onClick={ openSideMenu }

```

```

        onClick={ openSideMenu }
      >
        <MenuOutlinedIcon />
      </IconButton>

      { /* <NextLink href="/" passHref >
        <Link underline="none" color="white">
          <Typography variant="h6">OpenJira</Typography>
        </Link>
      </NextLink> */ }

      <Link href="/" underline="none" color="white">
        <Typography variant="h6">OpenJira</Typography>
      </Link>

    </Toolbar>
  </AppBar>

);
};

```

NewEntry.tsx

```

import { Box, Button, TextField } from "@mui/material";

import SaveOutlinedIcon from '@mui/icons-material/SaveOutlined';
import AddCircleOutlineOutlinedIcon from '@mui/icons-material/AddCircleOutlineOutlined';
import { ChangeEvent, useContext, useState } from "react";
import { EntriesContext } from "../../context/entries";
import { UIContext } from "../../context/ui";

export const NewEntry = () => {

  const { addNewEntry } = useContext( EntriesContext )

  const { isAddingEntry, setIsAddingEntry } = useContext( UIContext )

  const [inputValue, setInputValue] = useState('')

  const [touched, setTouched] = useState(false)

  const onTextFieldChanges = (event: ChangeEvent<HTMLInputElement>) => {
    setInputValue( event.target.value )
  }

  const onSave = () => {
    if ( inputValue.length === 0 ) return;

    addNewEntry(inputValue);
    setIsAddingEntry( false );
    setTouched( false );
    setInputValue('');
  }

  return (
    <Box sx={{ marginBottom: 2, paddingX: 1 }}>

      {

```

```

        isAddingEntry ? (
          <>
            < TextField
              fullWidth
              sx={{ marginTop: 2, marginBottom: 1 }}
              placeholder='Nueva Entrada'
              autoFocus
              multiline
              label='Nueva Entrada'
              helperText={ inputValue.length <= 0
                && touched && 'Ingrese un valor' }
              error={ inputValue.length <= 0 && touched }
              value={ inputValue }
              onChange={ onTextFieldChanges }
              onBlur={ () => setTouched( true ) }
            />

            <Box display='flex' justifyContent='space-between' >
              <Button
                variant='text'
                onClick={() => setIsAddingEntry( false ) }
              >
                Cancelar
              </Button>

              <Button
                variant='outlined'
                color='secondary'
                endIcon={ <SaveOutlinedIcon/> }
                onClick={ onSave }
              >
                Guardar
              </Button>
            </Box>
          </>
        ) : (
          <Button
            startIcon={ <AddCircleOutlineOutlinedIcon /> }
            fullWidth
            variant='outlined'
            onClick={() => setIsAddingEntry( true ) }
          >
            Agregar Tarea
          </Button>
        )
      }
    </Box>
  );
};

```

Sidebar.tsx

```

import { useContext } from "react";

import { Box, Divider, Drawer, List, ListItem, ListItemIcon, ListItemText, Typography } from "@mui/material"
import InboxOutlinedIcon from '@mui/icons-material/InboxOutlined';
import MailOutlineOutlinedIcon from '@mui/icons-material/MailOutlineOutlined';

```

```

import { UIContext } from "../../context/ui";

const menuItems: string[] = ['Inbox', 'Starred', 'Send Email', 'Drafts']

export const Sidebar = () => {

  const { sidemenuOpen, closeSideMenu } = useContext( UIContext )

  return (
    <Drawer
      anchor="left"
      open={ sidemenuOpen }
      onClose={ closeSideMenu }
    >
      <Box sx={{ width: 250 }}>
        <Box sx={{ padding: '5px 10px'}}>
          <Typography variant="h4"> Menú</Typography>
        </Box>

        <List>
          {
            menuItems.map( (text, index) => (
              <ListItem button key={ text }>
                <ListItemIcon>
                  { index % 2 ? <InboxOutlinedIcon/>
                    : <MailOutlineOutlinedIcon />}
                </ListItemIcon>
                <ListItemText primary={ text }/>
              </ListItem>
            ))
          }
        </List>

        <Divider/>

        <List>
          {
            menuItems.map( (text, index) => (
              <ListItem button key={ text }>
                <ListItemIcon>
                  { index % 2 ? <InboxOutlinedIcon/>
                    : <MailOutlineOutlinedIcon />}
                </ListItemIcon>
                <ListItemText primary={ text }/>
              </ListItem>
            ))
          }
        </List>
      </Box>
    </Drawer>
  )
}

```