

Crea tu cuenta

1. Datos personales

Nombre(s) \*

Apellido(s) \*

Correo electrónico \*

País \* ES Número de celular \*

Contraseña \*

Confirmación \*

☐ Mostrar contraseña

Ayúdame a crear un proyecto de react para hacer un formulario que reciba un soporte, necesito que los botones y demás componentes sean estéticos y bonitos, manejando un concepto de diseño similar a este



Para crear un formulario en React con un diseño estético similar al que has mostrado, podemos usar la biblioteca de componentes Material-UI, que proporciona una amplia gama de componentes estilizados y configurables.

Aquí tienes una guía paso a paso para crear un proyecto de React que incluye un formulario similar al de la imagen:

#### 1. Instalación de Create React App:

```
bash Copiar código
npx create-react-app my-form-project
cd my-form-project
```

#### 2. Instalación de Material-UI:

```
bash Copiar código
npm install @mui/material @emotion/react @emotion/styled
```

### 3. Creación del Componente del Formulario:

Crea un archivo `Form.js` dentro de la carpeta `src` y añade el siguiente código:

```
jsx Copiar código

// src/Form.js
import React from 'react';
import { Container, TextField, Button, MenuItem, Box, Typography } from '@mui/material';

const Form = () => {
  return (
    <Container maxWidth="sm">
      <Box
        component="form"
        sx={{
          display: 'flex',
          flexDirection: 'column',
          gap: '16px',
          marginTop: '40px',
          padding: '24px',
          borderRadius: '8px',
          boxShadow: '0 2px 10px rgba(0,0,0,0.1)',
          backgroundColor: '#fff',
        }}
      >
        <Typography variant="h5" component="h1">
          Crea tu cuenta
        </Typography>
        <TextField label="Nombres" variant="outlined" required />
        <TextField label="Apellidos" variant="outlined" required />
        <TextField label="Correo corporativo" variant="outlined" required />
        <Box sx={{ display: 'flex', alignItems: 'center', gap: '8px' }}>
          <TextField select label="Cód." value="57" variant="outlined">
            <MenuItem value="57">+57</MenuItem>
            { /* Agrega más códigos de país aquí */ }
          </TextField>
          <TextField label="Número de celular" variant="outlined" required />
        </Box>
        <TextField label="Contraseña" variant="outlined" type="password" />
        <TextField label="Confirmación" variant="outlined" type="password" />
        <Button variant="contained" color="primary" size="large">
          Siguiente
        </Button>
      </Box>
    </Container>
  );
}

export default Form;
```

#### 4. Uso del Componente en la Aplicación:

Modifica el archivo `App.js` para incluir el componente `Form`:

```
jsx Copiar código

// src/App.js
import React from 'react';
import Form from './Form';
import { CssBaseline, ThemeProvider, createTheme } from '@mui/material';

const theme = createTheme({
  palette: {
    background: {
      default: '#f0f2f5',
    },
  },
});

const App = () => {
  return (
    <ThemeProvider theme={theme}>
      <CssBaseline />
      <Form />
    </ThemeProvider>
  );
}

export default App;
```

#### 5. Estilización Adicional (Opcional):

Si deseas agregar más estilos personalizados, puedes crear un archivo CSS y aplicarlo según tus necesidades.

Finalmente, ejecuta tu proyecto:

```
bash Copiar código

npm start
```

Este es un ejemplo básico que puedes expandir y modificar según tus necesidades específicas. Material-UI es muy flexible y permite personalizar los componentes ampliamente.

es con react ts



Para crear un formulario con React y TypeScript utilizando Material-UI, sigue estos pasos:

### 1. Crear un proyecto con Create React App:

```
bash
npx create-react-app my-form-project --template typescript
cd my-form-project
```

### 2. Instalar Material-UI:

```
bash
npm install @mui/material @emotion/react @emotion/styled
```

### 3. Crear el componente del formulario:

Crea un archivo `Form.tsx` dentro de la carpeta `src` y añade el siguiente código:

```
tsx
// src/Form.tsx
import React, { useState } from 'react';
import { Container, TextField, Button, MenuItem, Box, Typography } from '@mui/m

const Form: React.FC = () => {
  const [firstName, setFirstName] = useState('');
  const [lastName, setLastName] = useState('');
  const [email, setEmail] = useState('');
  const [countryCode, setCountryCode] = useState('57');
  const [phoneNumber, setPhoneNumber] = useState('');
  const [password, setPassword] = useState('');
  const [confirmPassword, setConfirmPassword] = useState('');

  const handleSubmit = (event: React.FormEvent<HTMLFormElement>) => {
    event.preventDefault();
    // Aquí puedes manejar el envío del formulario
  };

  return (
    <Container maxWidth="sm">
      <Box
        component="form"
        onSubmit={handleSubmit}
        sx={{
          display: 'flex',
          flexDirection: 'column',
          gap: '16px',
          marginTop: '40px',
        }}
      >
```

```

        backgroundColor: '#fff',
      }}
    >
    <Typography variant="h5" component="h1">
      Crea tu cuenta
    </Typography>
    <TextField
      label="Nombres"
      variant="outlined"
      required
      value={firstName}
      onChange={(e) => setFirstName(e.target.value)}
    />
    <TextField
      label="Apellidos"
      variant="outlined"
      required
      value={lastName}
      onChange={(e) => setLastName(e.target.value)}
    />
    <TextField
      label="Correo corporativo"
      variant="outlined"
      required
      type="email"
      value={email}
      onChange={(e) => setEmail(e.target.value)}
    />
    <Box sx={{ display: 'flex', alignItems: 'center', gap: '8px' }}>
      <TextField
        select
        label="Cód."
        value={countryCode}
        variant="outlined"
        onChange={(e) => setCountryCode(e.target.value)}
        sx={{ maxWidth: '80px' }}
      >
        <MenuItem value="57">+57</MenuItem>
        {/* Agrega más códigos de país aquí */}
      </TextField>
      <TextField
        label="Número de celular"
        variant="outlined"
        required
        type="tel"
        value={phoneNumber}
        onChange={(e) => setPhoneNumber(e.target.value)}
        fullWidth
      />
    </Box>

```

```
    </Box>
    <TextField
      label="Contraseña"
      variant="outlined"
      type="password"
      required
      value={password}
      onChange={(e) => setPassword(e.target.value)}
    />
    <TextField
      label="Confirmación"
      variant="outlined"
      type="password"
      required
      value={confirmPassword}
      onChange={(e) => setConfirmPassword(e.target.value)}
    />
    <Button variant="contained" color="primary" size="large" type="button">
      Siguiente
    </Button>
  </Box>
</Container>
);
};

export default Form;
```

#### 4. Uso del componente en la aplicación:

Modifica el archivo `App.tsx` para incluir el componente `Form`:

```
tsx Copiar código

// src/App.tsx
import React from 'react';
import Form from './Form';
import { CssBaseline, ThemeProvider, createTheme } from '@mui/material';

const theme = createTheme({
  palette: {
    background: {
      default: '#f0f2f5',
    },
  },
});

const App: React.FC = () => {
  return (
    <ThemeProvider theme={theme}>
      <CssBaseline />
      <Form />
    </ThemeProvider>
  );
};

export default App;
```

#### 5. Ejecutar tu proyecto:

```
bash Copiar código

npm start
```

Este proyecto utiliza TypeScript junto con Material-UI para crear un formulario similar al de tu imagen. Puedes personalizar aún más los estilos y la funcionalidad según tus necesidades.



```

ayudame a crearle un componente para seleccionar el level risk "//
src/Form.tsx
import React, { useState } from 'react';
import { Container, TextField, Button, MenuItem, Box, Typography }
from '@mui/material';

const Form: React.FC = () => {
  const [title, setTitle] = useState("");
  const [description, setDescription] = useState("");
  const [status, setStatus] = useState('Open');

  const handleSubmit = (event:
React.FormEvent<HTMLFormElement>) => {
    event.preventDefault();
    // Aquí puedes manejar el envío del formulario
  };

  return (
    <Container maxWidth="sm">
      <Box
        component="form"
        onSubmit={handleSubmit}
        sx={{
          display: 'flex',
          flexDirection: 'column',
          gap: '16px',
          marginTop: '40px',
          padding: '24px',
          borderRadius: '8px',
          boxShadow: '0 2px 10px rgba(0,0,0,0.1)',
          backgroundColor: '#fff',
        }}
      >
        <Typography
          variant="h5"
          component="h1"
          sx={{
            margin: 0,
            fontFamily: 'Epilogue, sans-serif',
            fontWeight: 700,
            lineHeight: '140%',
            fontSize: '1.125rem',
            marginBottom: '32px',
            color: '#0A0F48',
          }}
        >
          Crea tu caso de soporte
        </Typography>
        <TextField
          label="Nombre del caso"
          variant="outlined"


```



```

    required
    value={title}
    onChange={(e) => setTitle(e.target.value)}
  />
  <TextField
    label="Compañía"
    variant="outlined"
    required
    value={title}
    onChange={(e) => setTitle(e.target.value)}
  />
  <TextField
    label="Descripción del caso"
    variant="outlined"
    required
    multiline
    rows={4}
    value={description}
    onChange={(e) => setDescription(e.target.value)}
  />
  <TextField
    select
    label="Estado"
    value={status}
    variant="outlined"
    onChange={(e) => setStatus(e.target.value)}
  >
    <MenuItem value="Open">Abierto</MenuItem>
    <MenuItem value="Closed">Cerrado</MenuItem>
    <MenuItem value="In Progress">En Progreso</MenuItem>
  </TextField>
  <Button
    variant="contained"
    color="primary"
    size="large"
    type="submit"
    sx={{
      backgroundColor: '#0C147B',
      '&:hover': { backgroundColor: '#080F5B' },
      borderRadius: '50px',
      textTransform: 'capitalize',
      color: 'fff'
    }}
  >
    Crear caso
  </Button>
</Box>
</Container>
);
};

```

export default For 



Para añadir un componente de selección de "level risk" (nivel de riesgo) en tu formulario, primero define un nuevo estado para el nivel de riesgo y luego añade un `TextField` con opciones de selección dentro de tu componente de formulario.

Aquí está tu componente actualizado con la selección de nivel de riesgo:

```
tsx Copiar código

import React, { useState } from 'react';
import { Container, TextField, Button, MenuItem, Box, Typography } from '@mui/material';

const Form: React.FC = () => {
  const [title, setTitle] = useState('');
  const [company, setCompany] = useState(''); // Nueva variable de estado para comp
  const [description, setDescription] = useState('');
  const [status, setStatus] = useState('Open');
  const [riskLevel, setRiskLevel] = useState('Low'); // Nueva variable de estado pa

  const handleSubmit = (event: React.FormEvent<HTMLFormElement>) => {
    event.preventDefault();
    // Aquí puedes manejar el envío del formulario
  };

  return (
    <Container maxWidth="sm">
      <Box
        component="form"
        onSubmit={handleSubmit}
        sx={{
          display: 'flex',
          flexDirection: 'column',
          gap: '16px',
          marginTop: '40px',
          padding: '24px',
          borderRadius: '8px',
          boxShadow: '0 2px 10px rgba(0,0,0,0.1)',
          backgroundColor: '#fff',
        }}
      >
        <Typography
          variant="h5"
          component="h1"
          sx={{
            margin: 0,
            fontFamily: 'Epilogue, sans-serif',
            fontWeight: 700,
            lineHeight: '140%',
            fontSize: '1.125rem',
            marginBottom: '32px',
            color: '#8A0FA8',
          }}
        />
      </Box>
    </Container>
  );
};
```

```

        marginBottom: '32px',
        color: '#0A0F48',
      }}
    >
    Crea tu caso de soporte
  </Typography>
  <TextField
    label="Nombre del caso"
    variant="outlined"
    required
    value={title}
    onChange={(e) => setTitle(e.target.value)}
  />
  <TextField
    label="Compañía"
    variant="outlined"
    required
    value={company} // Usar la variable de estado company
    onChange={(e) => setCompany(e.target.value)} // Actualizar la variable de
  />
  <TextField
    label="Descripción del caso"
    variant="outlined"
    required
    multiline
    rows={4}
    value={description}
    onChange={(e) => setDescription(e.target.value)}
  />
  <TextField
    select
    label="Estado"
    value={status}
    variant="outlined"
    onChange={(e) => setStatus(e.target.value)}
  >
    <MenuItem value="Open">Abierto</MenuItem>
    <MenuItem value="Closed">Cerrado</MenuItem>
    <MenuItem value="In Progress">En Progreso</MenuItem>
  </TextField>
  <TextField
    select
    label="Nivel de Riesgo"
    value={riskLevel}
    variant="outlined"
    onChange={(e) => setRiskLevel(e.target.value)}
  >

```

```
<Button
  variant="contained"
  color="primary"
  size="large"
  type="submit"
  sx={{
    backgroundColor: '#0C147B',
    '&:hover': { backgroundColor: '#008F5B' },
    borderRadius: '50px',
    textTransform: 'capitalize',
    color: '#fff'
  }}
>
  Crear caso
</Button>
</Box>
</Container>
);
};

export default Form;
```

Este código añade una nueva variable de estado `riskLevel` y un `TextField` con opciones de selección para "Bajo", "Medio", y "Alto" como niveles de riesgo. También se ha corregido el campo "Compañía" para usar una variable de estado separada.

ahora ayudame a crear una tabla con los casos de soportes inscritos

Para crear una tabla que muestre los casos de soporte inscritos, necesitas gestionar un estado en el componente principal que mantenga una lista de casos de soporte. También necesitas un componente de tabla que renderice esta lista.

Primero, actualiza tu formulario para añadir los casos de soporte a una lista cuando se envíe el formulario:

```
tsx Copiar código

import React, { useState } from 'react';
import { Container, TextField, Button, MenuItem, Box, Typography } from '@mui/material';

interface SupportCase {
  title: string;
  company: string;
  description: string;
  status: string;
  riskLevel: string;
}

const Form: React.FC<{ addSupportCase: (newCase: SupportCase) => void }> = ({ addSupportCase }) => {
  const [title, setTitle] = useState('');
  const [company, setCompany] = useState('');
  const [description, setDescription] = useState('');
  const [status, setStatus] = useState('Open');
  const [riskLevel, setRiskLevel] = useState('Low');

  const handleSubmit = (event: React.FormEvent<HTMLFormElement>) => {
    event.preventDefault();
    const newCase: SupportCase = {
      title,
      company,
      description,
      status,
      riskLevel,
    };
    addSupportCase(newCase);
  };

  return (
    <Container>
      <Box>
        <TextField>
          <label>Title</label>
          <input type="text" value={title} onChange={e => setTitle(e.target.value)} />
        </TextField>
        <TextField>
          <label>Company</label>
          <input type="text" value={company} onChange={e => setCompany(e.target.value)} />
        </TextField>
        <TextField>
          <label>Description</label>
          <input type="text" value={description} onChange={e => setDescription(e.target.value)} />
        </TextField>
        <TextField>
          <label>Status</label>
          <input type="text" value={status} onChange={e => setStatus(e.target.value)} />
        </TextField>
        <TextField>
          <label>Risk Level</label>
          <input type="text" value={riskLevel} onChange={e => setRiskLevel(e.target.value)} />
        </TextField>
        <Button type="button" onClick={handleSubmit}>Add Case</Button>
      </Box>
    </Container>
  );
};
```

```

return (
  <Container maxWidth="sm">
    <Box
      component="form"
      onSubmit={handleSubmit}
      sx={{
        display: 'flex',
        flexDirection: 'column',
        gap: '16px',
        marginTop: '40px',
        padding: '24px',
        borderRadius: '8px',
        boxShadow: '0 2px 10px rgba(0,0,0,0.1)',
        backgroundColor: '#fff',
      }}
    >
      <Typography
        variant="h5"
        component="h1"
        sx={{
          margin: 0,
          fontFamily: 'Epilogue, sans-serif',
          fontWeight: 700,
          lineHeight: '140%',
          fontSize: '1.125rem',
          marginBottom: '32px',
          color: '#0A0F48',
        }}
      >
        Crea tu caso de soporte
      </Typography>
      <TextField
        label="Nombre del caso"
        variant="outlined"
        required
        value={title}
        onChange={(e) => setTitle(e.target.value)}
      />
    </Container>
  )

```

Ahora, crea un componente para la tabla que muestre los casos de soporte:

```
tsx Copiar código

import React from 'react';
import { Container, Table, TableBody, TableCell, TableContainer, TableHead, TableRow } from '@mui/material';

interface SupportCase {
  title: string;
  company: string;
  description: string;
  status: string;
  riskLevel: string;
}

const SupportCaseTable: React.FC<{ supportCases: SupportCase[] }> = ({ supportCases }) => {
  return (
    <Container>
      <TableContainer component={Paper} sx={{ marginTop: '40px' }}>
        <Table>
          <TableHead>
            <TableRow>
              <TableCell>Titulo</TableCell>
              <TableCell>Compañía</TableCell>
              <TableCell>Descripción</TableCell>
              <TableCell>Estado</TableCell>
              <TableCell>Nivel de Riesgo</TableCell>
            </TableRow>
          </TableHead>
          <TableBody>
            {supportCases.map((supportCase, index) => (
              <TableRow key={index}>
                <TableCell>{supportCase.title}</TableCell>
                <TableCell>{supportCase.company}</TableCell>
                <TableCell>{supportCase.description}</TableCell>
                <TableCell>{supportCase.status}</TableCell>
                <TableCell>{supportCase.riskLevel}</TableCell>
              </TableRow>
            ))}
          </TableBody>
        </Table>
      </TableContainer>
    </Container>
  );
};
```

Finalmente, integra ambos componentes en el componente principal:

```
tsx Copiar código

import React, { useState } from 'react';
import Form from './Form';
import SupportCaseTable from './SupportCaseTable';

interface SupportCase {
  title: string;
  company: string;
  description: string;
  status: string;
  riskLevel: string;
}

const App: React.FC = () => {
  const [supportCases, setSupportCases] = useState<SupportCase[]>([]);

  const addSupportCase = (newCase: SupportCase) => {
    setSupportCases([...supportCases, newCase]);
  };

  return (
    <div>
      <Form addSupportCase={addSupportCase} />
      <SupportCaseTable supportCases={supportCases} />
    </div>
  );
};

export default App;
```

En este código, el componente `App` gestiona el estado de los casos de soporte y se asegura de pasar la función `addSupportCase` al componente `Form` para añadir nuevos casos. El componente `SupportCaseTable` recibe la lista de casos de soporte y los renderiza en una tabla.

