

# Manual Técnico

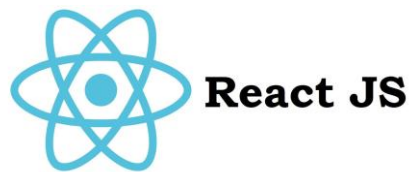
## Herramientas Utilizadas

NodeJs v16.13.1



<https://nodejs.org>

ReactJs v18.2.0



<https://es.reactjs.org/>

Microsoft SQL Server 2015



<https://www.microsoft.com/es-es/sql-server/sql-server-downloads>

Microsoft SQL Server Management Studio 2017



<https://www.microsoft.com>

## Visual Studio Code



Visual Studio Code

<https://netbeans.org/downloads/8.2/>

## FrontEnd - ReactJS

### Paquetes Implementados

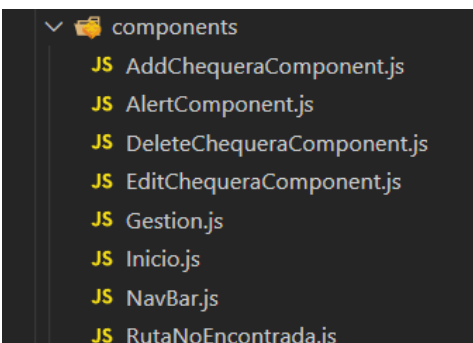
```
"@emotion/react": "^11.10.5",
"@emotion/styled": "^11.10.5",
"@fontsource/roboto": "^4.5.8",
"@mui/icons-material": "^5.10.15",
"@mui/material": "^5.10.15",
"@testing-library/jest-dom": "^5.16.5",
"@testing-library/react": "^13.4.0",
"@testing-library/user-event": "^13.5.0",
"axios": "^1.2.0",
"react": "^18.2.0",
"react-dom": "^18.2.0",
"react-router-dom": "^6.4.3",
"react-scripts": "5.0.1",
```

### Rutas Implementadas

```
<div className="App">
  <NavBar/>
  <Router>
    <Routes>
      <Route path="/" element={<Inicio/>} />
      <Route path="/Inicio" element={<Inicio/>} />
      <Route path="/Gestion" element={<Gestion/>} />
      <Route path="/*" element={<RutaNoEncontrada/>} />
    </Routes>
  </Router>
</div>
```

## Componentes

- AddChequeraComponent: Componente dialog que permitirá agregar chequeras en base a un usuario y/o cuenta de banco.
- DeleteChequeraComponent: Componente dialog que permitirá eliminar chequeras que se encuentren en el estado de “En proceso”.
- EditChequeraComponent: Componente dialog que permitirá eliminar chequeras que se encuentren en el estado de “En proceso”.
- Gestion: Componente dialog que permitirá eliminar chequeras que se encuentren en el estado de “En proceso”.
- AlertComponent: Componente para gestionar los mensajes de alert del sistema.



## AddChequeraComponent

### Hooks para para el manejo del componente autocomplete

```
//##### HOOKS #####  
const { open, setOpen, alertFuncion, resetSearch } = props;  
const [usuarioList, setUsuarioList] = useState([]);  
const [usuarioInputValue, setUsuarioInputValue] = React.useState("");  
const [selectedUsuario, setSelectedUsuario] = React.useState({});  
const [cuentaList, setCuentaList] = useState([]);  
const [cuentaInputValue, setCuentaInputValue] = React.useState("");  
const [selectedCuenta, setSelectedCuenta] = React.useState({});
```

### Hook para actualizar la lista de usuario cuando se escriba en el componente de autocomplete de usuarios

```
useEffect(() => {  
  setSelectedCuenta({});  
  setCuentaList([]);  
  cuentaService  
    .getByNombreDpi({ idUsuario: selectedUsuario?.id })  
    .then((response) => {  
      if (response?.valid) {  
        setCuentaList(response.data);  
      }  
    })  
    .catch((error) => {  
      console.error(error);  
    });  
}, [selectedUsuario]);
```

```

<Autocomplete
  id="idSearchNombre"
  freeSolo
  inputValue={usuarioInputValue}
  onInputChange={(event, searchValue) => {
    setUsuarioInputValue(searchValue);
  }}
  onChange={(event, newValue) => {
    setSelectedUsuario(newValue);
  }}
  options={
    usuariolist?.map((usuario) => ({
      id: usuario.idUsuario,
      label:
        usuario.nombre + " " + usuario.apellido + " - " + usuario.dpi,
    })) || []
  }
  renderInput={(params) => <TextField {...params} label="Usuario" />}
/>

```

**Agregar Chequera**

Seleccionar Usuario

Usuario

ma

María Ivet Montilla Fraga - 2309494875371

Anna Maria Ubeda Llorente - 6223786713347

Esteban Mario Montoya Martínez - 9901567041884

AGREGAR CANCELAR

Hook para actualizar la lista de cuentas bancarias cuando se escriba en el componente de autocomplete de cuenta bancaria o se seleccione un usuario.

```

useEffect(() => {
  setSelectedCuenta({});
  setCuentaList([]);
  cuentaService
    .getByNombreDpi({ idUsuario: selectedUsuario?.id })
    .then((response) => {
      if (response?.valid) {
        setCuentaList(response.data);
      }
    })
    .catch((error) => {
      console.error(error);
    });
}, [selectedUsuario]);

```

```

<Autocomplete
  id="idSearchCuenta"
  freeSolo
  inputValue={cuentaInputValue}
  onChange={(event, searchValue) => {
    setCuentaInputValue(searchValue);
  }}
  onInputChange={(event, newValue) => {
    setSelectedCuenta(newValue);
  }}
  options={
    cuentaList?.map((cuenta) => ({
      id: cuenta.idCuenta,
      label:
        cuenta.idCuenta + " - " + cuenta?.tipoCuentaModel?.detalle,
    })) || []
  }
  renderInput={(params) => (
    <TextField {...params} label="Cuenta Bancaria" />
  )}
/>

```

## Agregar Chequera

Seleccionar Usuario

Usuario

María Ivet Montilla Fraga - 2309494875371

Seleccionar Cuenta Bancaria

Cuenta Bancaria

×

1 - Ahorro

15 - Ahorro

Función para guardar la chequera

```

##### FUNCIONES #####
const agregarChequera = () => {
  if(!selectedCuenta?.id){
    alertFuncion(true,true,"seleccione una cuenta");
    return;
  }
  chequeraService
    .addChequera({ idCuenta: selectedCuenta?.id })
    .then((response) => {
      if (response?.valid) {
        alertFuncion(true,false,"Asignada con exito");
        resetSearch();
        setOpen(false);
      } else {
        alertFuncion(true,true,"Error al asignar");
      }
    })
    .catch((error) => {
      console.error(error);
      alertFuncion(true,true,"Error al asignar");
    });
};

const handleClose = () => {
  setOpen(false);
};

```

Componente que utiliza los componentes de Snackar y Alert (Material UI) para mostrar Alerta temporales

```
export default function AlertMessage(props) {
  const { open, handleClose, message, messageError } = props;
  const vertical = "top";
  const horizontal = "right";
  return (
    <div>
      <Snackbar
        anchorOrigin={{ vertical, horizontal }}
        open={open}
        autoHideDuration={6000}
        onClose={handleClose}
        key={vertical + horizontal}
      >
        <Alert onClose={handleClose} severity={!messageError?"success":"error"} sx={{ width: "100%" }}>
          {message}
        </Alert>
      </Snackbar>
    </div>
  );
}
```

INICIO GESTIÓN DE CHEQUERAS

Asignada con éxito

Lista de Chequeras

## DeleteChequeraComponent







Función para eliminar la chequera

```
//##### FUNCIONES #####
const eliminarChequera = () => {
  chequeraService
    .deleteChequera({
      idChequera: chequeraActual.idChequera,
    })
    .then((response) => {
      if (response?.valid) {
        alertFuncion(true, false, "Eliminada con éxito");
        resetSearch();
        setOpen(false);
      } else {
        alertFuncion(true, true, "Error al asignar");
      }
    })
    .catch((error) => {
      console.error(error);
      alertFuncion(true, true, "Error al asignar");
    });
};
```

Ventana para aceptar o cancelar la eliminación de una chequera

```
<div>
  <Dialog
    open={open}
    onClose={handleClose}
    aria-labelledby="alert-dialog-title"
    aria-describedby="alert-dialog-description"
  >
    <DialogTitle id="alert-dialog-title">
      {"¿Desea eliminar la Chequera: " + chequeraActual.idChequera + " ?"}
    </DialogTitle>
    <DialogContent></DialogContent>
    <DialogActions>
      <Button onClick={eliminarChequera}>Eliminar</Button>
      <Button onClick={handleClose} autoFocus>
        Volver
      </Button>
    </DialogActions>
  </Dialog>
</div>
```

### Lista de Chequeras

Chequera	Fecha de Creación	Estado	Cuenta	Titular	Acciones
1	26-11-2022 19:21:42	Entregada	5 - Ahorro	Florencia Sabela Tortosa de la Cruz - 6130904240740	
2	26-11-2022 19:24:19	Cancelada	2 - Ahorros	Felipe González López Rodríguez - 851405748890	
3	26-11-2022 20:47:31	Cancelada	2 - Ahorros	Felipe González López Rodríguez - 851405748890	
4	26-11-2022 21:36:01	En Proceso	11 - Ahorro	Anna Maria Ubéda Llorente - 6223786713347	  



## EditChequeraComponent

Hooks para el manejo de dialog y select

```
✓ export default function AddChequeraComponent(props) {  
  //##### HOOKS #####  
  const { open, setOpen, alertFuncion, chequeraActual, resetSearch } = props;  
  
  const [selectedEstado, setSelectedEstado] = React.useState(1);  
  
  ✓ useEffect(() => {  
    if(chequeraActual?.estadoChequeraModel?.idEstado)  
      setSelectedEstado(chequeraActual?.estadoChequeraModel?.idEstado);  
  }, [chequeraActual]);  
}
```

Dialog para editar la chequera

```
>  
  <DialogTitle>Editar Chequera</DialogTitle>  
  <DialogContent>  
    <Typography variant="h6" color="initial">  
      Codigo de Chequera: {chequeraActual?.idChequera}  
    </Typography>  
  
    <Box sx={{ minWidth: 120 }}>  
      <FormControl fullWidth>  
        <InputLabel variant="standard" htmlFor="uncontrolled-native">  
          Estado Chequera  
        </InputLabel>  
        <Select  
          value={selectedEstado}  
          onChange={(event) => {  
            setSelectedEstado(event.target.value);  
          }}  
        >  
          <MenuItem key={1} value={1}>  
            En Proceso  
          </MenuItem>  
          <MenuItem key={2} value={2}>  
            Entregada  
          </MenuItem>  
        </Select>  
      </FormControl>  
    </Box>  
  </DialogContent>
```

INICIO GESTIÓN DE CHEQUERAS

## Lista de Chequeras +

Chequera	Fecha de Creación	Estado	Cuenta	Titular	Acciones
1	26-11-2022 19:21:42	Entregada	5 - Ahorro	Florencia SabelaTortosa de la Cruz - 6130904240740	<span>⋮</span>
2	26-11-2022 19:24:19	Cancelada		748890	<span>⋮</span>
3	26-11-2022 20:47:31	Cancelada		748890	<span>⋮</span>
4	26-11-2022 21:36:01	En Proceso		713347	<span>✎</span> <span>🗑</span> <span>⋮</span>

**Editar Chequera**

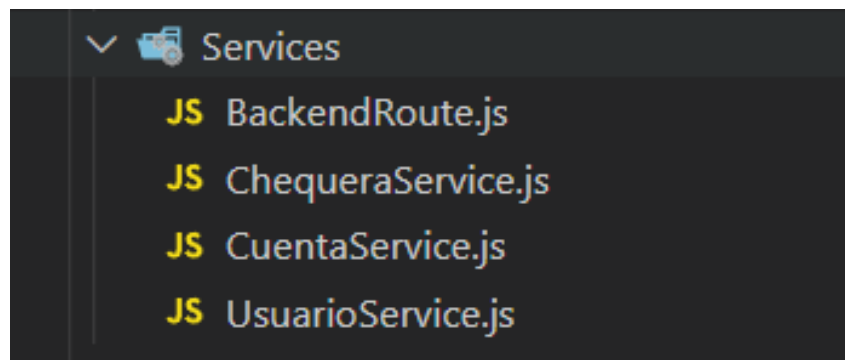
**Codigo de Chequera: 4**

Estado Chequera

En Proceso

EDITAR CANCELAR

## Servicios



### BackendRoute:

Manejo de las urls del sistema.

```
const BackendRoute = {
  BackendBaseUrl: "http://localhost:8080",

  ChequeraRoute: {
    get: "/chequera/get",
    add: "/chequera/add",
    edit: "/chequera/editar",
    delete: "/chequera/delete"
  },

  UsuarioRoute: {
    getByNombreDpi: "/usuario/getByNombreDpi",
  },

  CuentaRoute: {
    getByUser: "/cuenta/getByUser",
  },
};

export { BackendRoute };
```

## ChequeraRoute:

Peticiones enfocadas en la tabla de Chequera

```
import { BackendRoute } from "../BackendRoute";
import { default as axios } from "axios";
const { BackendBaseUrl, ChequeraRoute } = BackendRoute;

export async function getChequera(objSearch) {
  try {
    const { data } = await axios.get(BackendBaseUrl + ChequeraRoute.get, {
      params: objSearch,
    });
    return data;
  } catch (error) {
    console.error(error);
    return {};
  }
}

export async function addChequera(objSearch) {
  try {
    const { data } = await axios.post(BackendBaseUrl + ChequeraRoute.add, {
      ...objSearch,
    });
    return data;
  } catch (error) {
    console.error(error);
    return {};
  }
}

export async function editarChequera(objSearch) {
  try {
    const { data } = await axios.put(BackendBaseUrl + ChequeraRoute.edit, {
      ...objSearch,
```

## CuentaRoute:

Peticiones enfocadas en la tabla de Cuenta Bancaria

```

import { BackendRoute } from "../BackendRoute";
import { default as axios } from "axios";
const { BackendBaseUrl, CuentaRoute } = BackendRoute;

export async function getByNombreDpi(objSearch) {
  try {
    const { data } = await axios.get(BackendBaseUrl + CuentaRoute.getByUser, {
      params: objSearch,
    });
    return data;
  } catch (error) {
    console.error(error);
    return {};
  }
}

```

## UsuarioRoute:

Peticiones enfocadas en la tabla de Usuario

```

export async function getByNombreDpi(objSearch) {
  try {
    const { data } = await axios.get(BackendBaseUrl + UsuarioRoute.getByNombreDpi, {
      params: objSearch,
    });
    return data;
  } catch (error) {
    console.error(error);
    return {};
  }
}

```

# Acciones Permitidas

Una chequera solo podrá ser editado o eliminada cuando aun se encuentra en el estado “En Proceso”

## Lista de Chequeras

Chequera	Fecha de Creación	Estado	Cuenta	Titular	Acciones
1	26-11-2022 19:21:42	Entregada	5 - Ahorro	Florencia Sabela Tortosa de la Cruz - 6130904240740	
2	26-11-2022 19:24:19	Cancelada	2 - Monetaria	Felix Candido Lucas Padilla - 4851405748890	
3	26-11-2022 20:47:31	Cancelada	2 - Monetaria	Felix Candido Lucas Padilla - 4851405748890	
4	26-11-2022 21:36:01	En Proceso	11 - Ahorro	Anna Maria Ubeda Llorente - 6223786713347	  

# BackEnd – Spring Boot

## Dependencias Utilizadas

```
<dependencies>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
  </dependency>

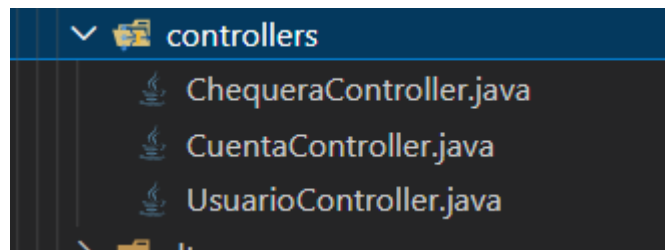
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-test</artifactId>
    <scope>test</scope>
  </dependency>

  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-devtools</artifactId>
    <scope>runtime</scope>
  </dependency>

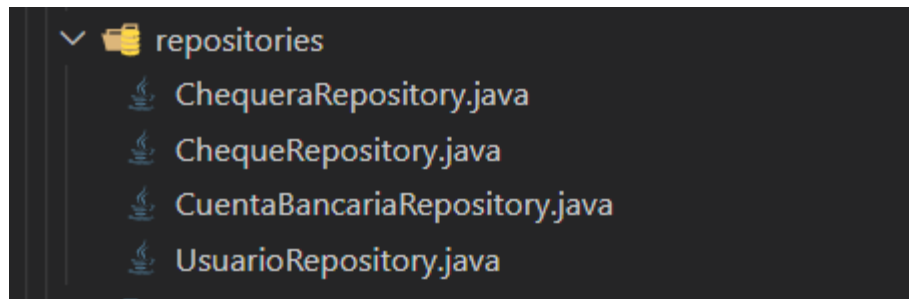
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-jpa</artifactId>
  </dependency>

  <dependency>
    <groupId>com.microsoft.sqlserver</groupId>
    <artifactId>mssql-jdbc</artifactId>
    <scope>runtime</scope>
  </dependency>
</dependencies>
```

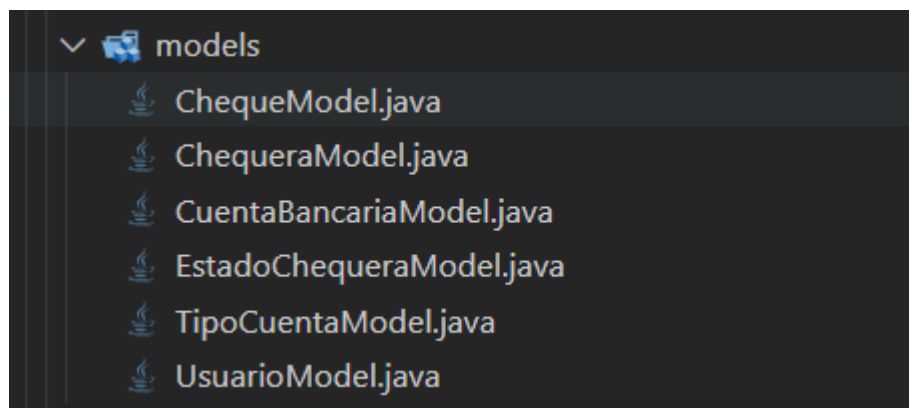
## Controllers



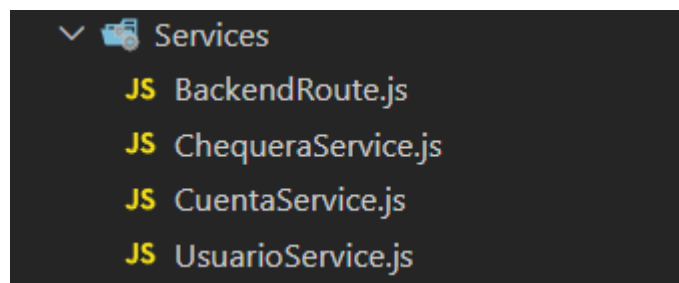
## Repositories



## Models



## Services



## ResponseMessage

Todas las peticiones retornaran un objeto de tipo ResponseMessage; en el cual se identifican los siguientes atributos:

- Object data: Toda información que sea enviada para sus posterior uso (select,update,etc)
- boolean Valid: Si la petición fue correcta o existió algún error.
- String msg: mensaje de error o petición correcta.

```
public class ResponseMessage {  
    private Object data;  
    private boolean valid;  
    private String msg;  
  
    public ResponseMessage(Object data, boolean valid, String msg) {  
        this.data = data;  
        this.valid = valid;  
        this.msg = msg;  
    }  
}
```

## Estados del Sistema

### Estados Chequera:

- En proceso: La chequera fue asignada a una cuenta.
- Eliminada: La chequera fue cancelada y/o eliminada
- Entregada: La chequera fue entregada al cliente.

