My Projects My Issues Explore

Tomaxido > INF225-2024-1-GRUPO-24 > main ✓

⚠ The last analysis h

Summary **Issues** Security Hotspots Measures Code Activity

2 / 25 issues

INF236-Users-Frontend-Boiler...

Unexpected empty source

INF236-Users-Frontend-Boiler...

Prefer using an optional chain expression instead, as it's more concise and easier to read.

Prefer using an optional chain expression instead, as it's more concise and easier to read.

Prefer using an optional chain expression instead, as it's more concise and easier to read.

Remove this useless assignment to variable "day".

Prefer using an optional chain expression instead, as it's more concise and easier to read.

Prefer using an optional chain expression instead, as it's more concise and easier to read.

Remove this useless assignment to variable "handleChange".

Do not use Array index in keys

INF236-Users-Frontend-Boiler...

Remove this useless assignment to variable "error".

Remove this useless assignment to variable "state".

INF236-Users-Frontend-Boiler...

Do not use Array index in keys

INF236-Users-Frontend-Boiler...

Remove this useless assignment to variable "error".

Intentionality | Not clear

Prefer using an optional chain expression instead, as it's more concise and easier to read.

Optional chaining should be preferred **javascript:S6582**

Software qualities impacted: Maintainability

○ Open Not assigned Code Smell Minor

Tags
No ta

Line
L37

Effor
5 mir

Intro
10 da

Where is the issue? | **Why is this an issue?** | **How can I fix it?** | **Activity** | **Mor**

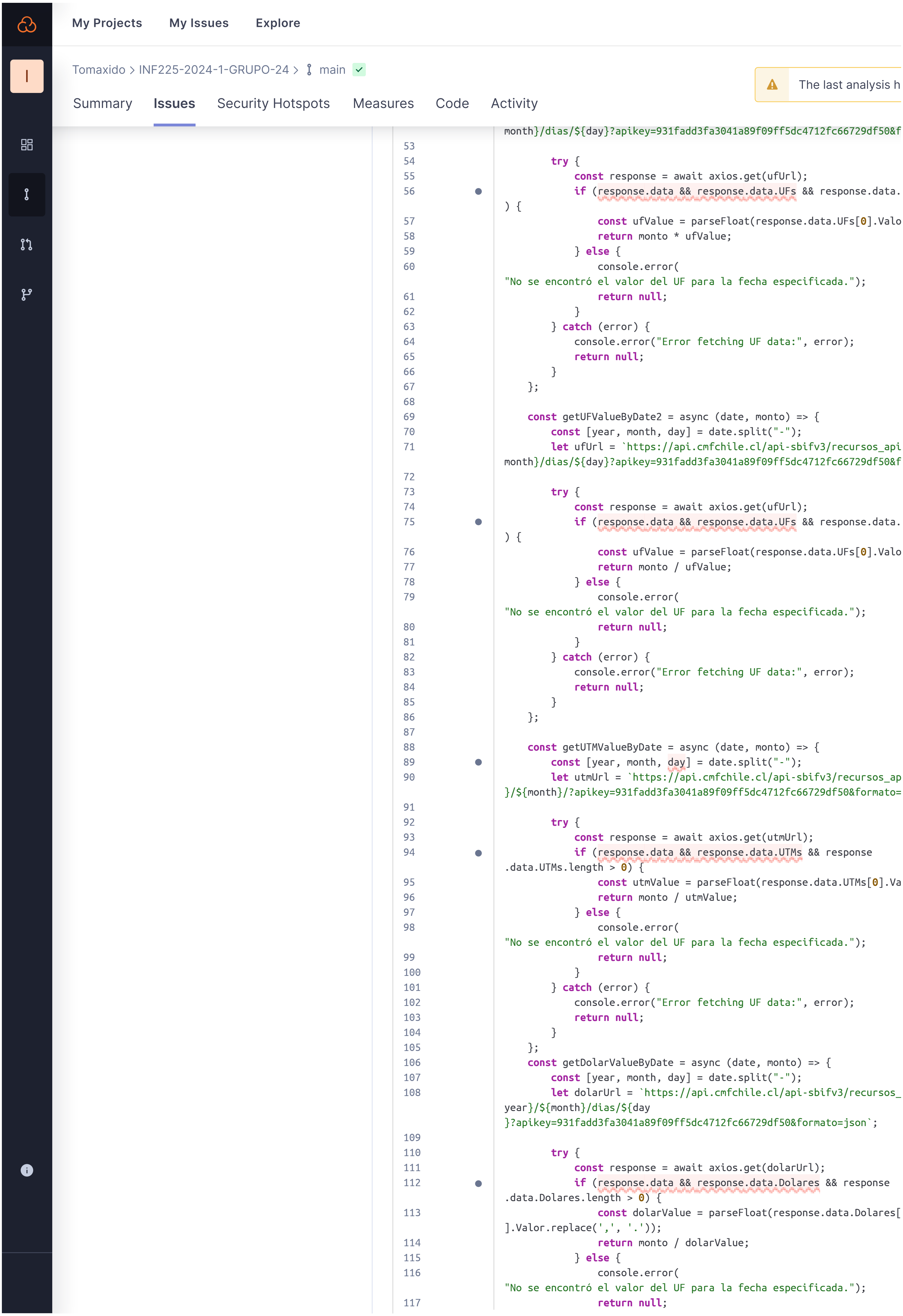
INF236-Users-Frontend-Boilerplate-master/src/views/conversion_cuotas/index.jsx

See all is

```
1   felipe…   import React, { useEffect, useState } from "react";
2             import useSWR from "swr";
3             import Cookies from 'universal-cookie';
4             import { getAllSolis } from '../../repositories/solicitud';
5             import Modal from 'react-modal';
6             import axios from 'axios';
7
8             Modal.setAppElement('#root'); // Configura el elemento raíz para acc
9
10            export default function Index() {
11                const cookies = new Cookies();
12                const cargoCookie = cookies.get('cargo');
13
14                const [state, setState] = useState({
15                    informeData: [],
16                });
17
18                const [modalIsOpen, setModalIsOpen] = useState(false);
19                const [selectedData, setSelectedData] = useState(null);
20                const [euroValueC, setEuroValueC] = useState(null);
21                const [clpValueC, setCLPValueC] = useState(null);
22                const [utmValueC, setUtmValueC] = useState(null);
23                const [dolarValueC, setDolarValueC] = useState(null);
24                const [euroValueM, setEuroValueM] = useState(null);
25                const [ufValueM, setUFValueM] = useState(null);
26                const [utmValueM, setUtmValueM] = useState(null);
27                const [dolarValueM, setDolarValueM] = useState(null);
28
29
30                const getEuroValueByDate = async (date, monto) => {
31                    const [year, month, day] = date.split("-");
32                    let euroUrl = `https://api.cmfchile.cl/api-sbifv3/recursos_a
              }/${month}/dias/${day
              }?apikey=931fadd3fa3041a89f09ff5dc4712fc66729df50&formato=json`;
33
34                    try {
35                        const response = await axios.get(euroUrl);
36                        console.log(response);
37                        if (response.data && response.data.Euros && response
              .data.Euros.length > 0) {
```

Prefer using an optional chain expression instead, as it's mo and easier to read.

```
38                            const euroValue = parseFloat(response.data.Euros[0].
              ',', '.'));
39                            return monto / euroValue;
40                        } else {
41                            console.error(
              "No se encontró el valor del euro para la fecha especificada.");
42                            return null;
43                        }
44                    } catch (error) {
```



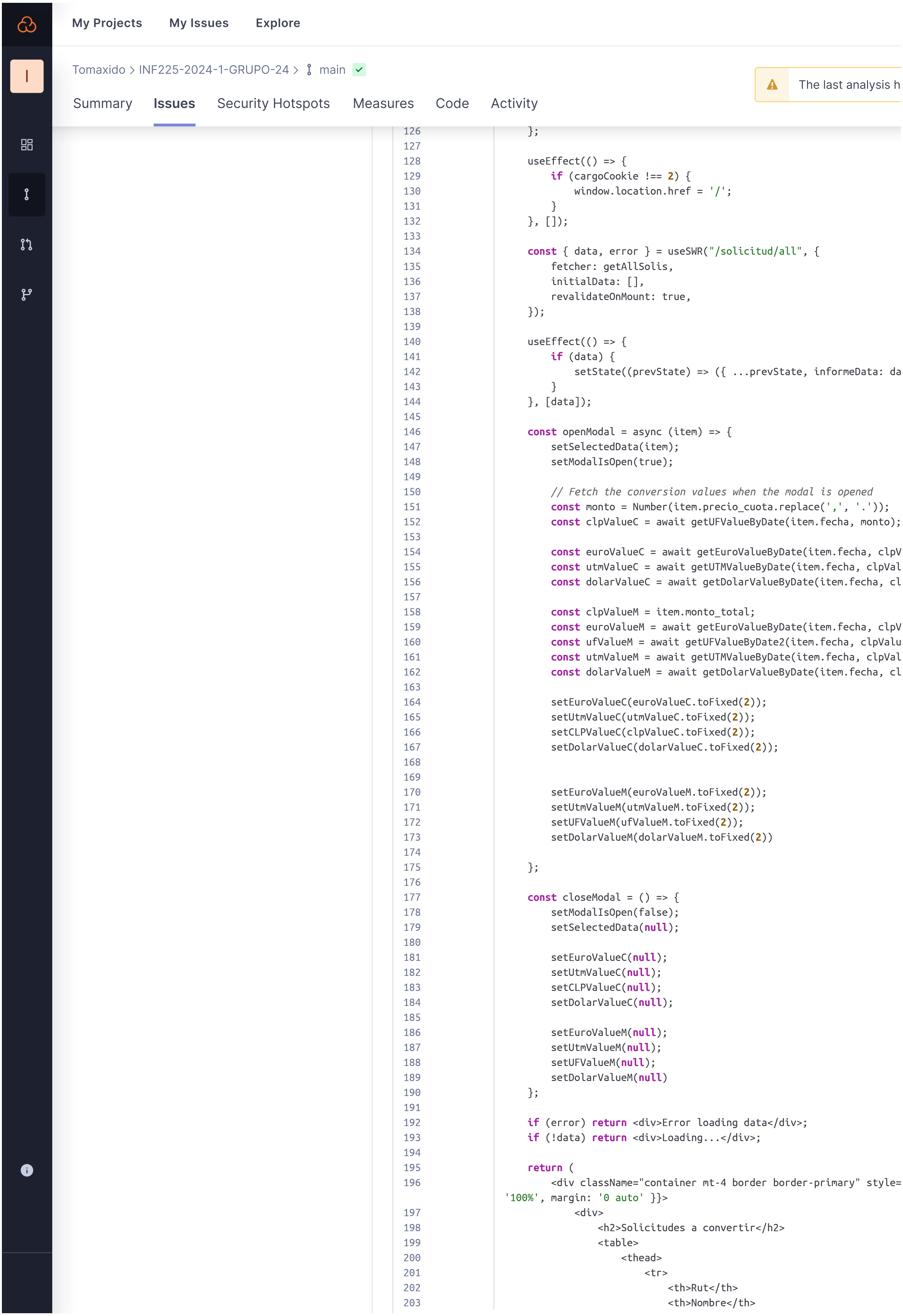
My Projects My Issues Explore

Tomaxido > INF225-2024-1-GRUPO-24 > main ✓

⚠ The last analysis h

Summary Issues Security Hotspots Measures Code Activity

```
month}/dias/${day}?apikey=931fadd3fa3041a89f09ff5dc4712fc66729df50&f

        try {
            const response = await axios.get(ufUrl);
            if (response.data && response.data.UFs && response.data.
) {
                const ufValue = parseFloat(response.data.UFs[0].Valo
                return monto * ufValue;
            } else {
                console.error(
"No se encontró el valor del UF para la fecha especificada.");
                return null;
            }
        } catch (error) {
            console.error("Error fetching UF data:", error);
            return null;
        }
    };

    const getUFValueByDate2 = async (date, monto) => {
        const [year, month, day] = date.split("-");
        let ufUrl = `https://api.cmfchile.cl/api-sbifv3/recursos_api
month}/dias/${day}?apikey=931fadd3fa3041a89f09ff5dc4712fc66729df50&f

        try {
            const response = await axios.get(ufUrl);
            if (response.data && response.data.UFs && response.data.
) {
                const ufValue = parseFloat(response.data.UFs[0].Valo
                return monto / ufValue;
            } else {
                console.error(
"No se encontró el valor del UF para la fecha especificada.");
                return null;
            }
        } catch (error) {
            console.error("Error fetching UF data:", error);
            return null;
        }
    };

    const getUTMValueByDate = async (date, monto) => {
        const [year, month, day] = date.split("-");
        let utmUrl = `https://api.cmfchile.cl/api-sbifv3/recursos_ap
}/${month}/?apikey=931fadd3fa3041a89f09ff5dc4712fc66729df50&formato=

        try {
            const response = await axios.get(utmUrl);
            if (response.data && response.data.UTMs && response
.data.UTMs.length > 0) {
                const utmValue = parseFloat(response.data.UTMs[0].Va
                return monto / utmValue;
            } else {
                console.error(
"No se encontró el valor del UF para la fecha especificada.");
                return null;
            }
        } catch (error) {
            console.error("Error fetching UF data:", error);
            return null;
        }
    };
    const getDolarValueByDate = async (date, monto) => {
        const [year, month, day] = date.split("-");
        let dolarUrl = `https://api.cmfchile.cl/api-sbifv3/recursos_
year}/${month}/dias/${day
}?apikey=931fadd3fa3041a89f09ff5dc4712fc66729df50&formato=json`;

        try {
            const response = await axios.get(dolarUrl);
            if (response.data && response.data.Dolares && response
.data.Dolares.length > 0) {
                const dolarValue = parseFloat(response.data.Dolares[
].Valor.replace(',', '.'));
                return monto / dolarValue;
            } else {
                console.error(
"No se encontró el valor del UF para la fecha especificada.");
                return null;
```



```
    };

    useEffect(() => {
        if (cargoCookie !== 2) {
            window.location.href = '/';
        }
    }, []);

    const { data, error } = useSWR("/solicitud/all", {
        fetcher: getAllSolis,
        initialData: [],
        revalidateOnMount: true,
    });

    useEffect(() => {
        if (data) {
            setState((prevState) => ({ ...prevState, informeData: da
        }
    }, [data]);

    const openModal = async (item) => {
        setSelectedData(item);
        setModalIsOpen(true);

        // Fetch the conversion values when the modal is opened
        const monto = Number(item.precio_cuota.replace(',', '.'));
        const clpValueC = await getUFValueByDate(item.fecha, monto);

        const euroValueC = await getEuroValueByDate(item.fecha, clpV
        const utmValueC = await getUTMValueByDate(item.fecha, clpVal
        const dolarValueC = await getDolarValueByDate(item.fecha, cl

        const clpValueM = item.monto_total;
        const euroValueM = await getEuroValueByDate(item.fecha, clpV
        const ufValueM = await getUFValueByDate2(item.fecha, clpValu
        const utmValueM = await getUTMValueByDate(item.fecha, clpVal
        const dolarValueM = await getDolarValueByDate(item.fecha, cl

        setEuroValueC(euroValueC.toFixed(2));
        setUtmValueC(utmValueC.toFixed(2));
        setCLPValueC(clpValueC.toFixed(2));
        setDolarValueC(dolarValueC.toFixed(2));


        setEuroValueM(euroValueM.toFixed(2));
        setUtmValueM(utmValueM.toFixed(2));
        setUFValueM(ufValueM.toFixed(2));
        setDolarValueM(dolarValueM.toFixed(2))

    };

    const closeModal = () => {
        setModalIsOpen(false);
        setSelectedData(null);

        setEuroValueC(null);
        setUtmValueC(null);
        setCLPValueC(null);
        setDolarValueC(null);

        setEuroValueM(null);
        setUtmValueM(null);
        setUFValueM(null);
        setDolarValueM(null)
    };

    if (error) return <div>Error loading data</div>;
    if (!data) return <div>Loading...</div>;

    return (
        <div className="container mt-4 border border-primary" style=
'100%', margin: '0 auto' }}>
            <div>
                <h2>Solicitudes a convertir</h2>
                <table>
                    <thead>
                        <tr>
                            <th>Rut</th>
                            <th>Nombre</th>
```

My Projects My Issues Explore

Tomaxido > INF225-2024-1-GRUPO-24 > main ✓

⚠ The last analysis h

Summary Issues Security Hotspots Measures Code Activity

```
                                    <td>{item.rut}</td>
                                    <td>{item.nombre}</td>
                                    <td>{item.fecha}</td>
                                    <td>
                                        <button className="btn btn-s
onClick=
{() => openModal(item)}>Ver</button>
                                    </td>
                                </tr>
                            );
                        })}
                    </tbody>
                </table>
            </div>

            <Modal
                isOpen={modalIsOpen}
                onRequestClose={closeModal}
                contentLabel="Detalles de la Solicitud"
                style={{
                    overlay: {
                        backgroundColor: 'rgba(0, 0, 0, 0.75)',
                    },
                    content: {
                        color: 'black',
                        top: '50%',
                        left: '50%',
                        right: '800%',
                        bottom: 'auto',
                        marginRight: '-50%',
                        transform: 'translate(-50%, -50%)',
                    },
                }}
            >
                {selectedData ? (
                    <div>
                        <h2>Conversiones</h2>
                        <div style={{ display: 'flex' }}>
                            <div style={{ flex: 1, paddingRight: '10
                                <p><strong>Nombre:</strong> {selecte
</p>
                                <p><strong>Fecha:</strong> {selected
</p>
                                <p><strong>Monto UF:</strong> {ufVal
                                <p><strong>Monto CLP:</strong> {sele
.monto_total}</p>
                                <p><strong>Monto UTM:</strong> {utmV
                                <p><strong>Monto Euro:</strong> {eur
                                <p><strong>Monto Dolar:</strong> {do
                            </div>
                            <div style={{ flex: 1, paddingLeft: '10p
                                <p><strong>RUT:</strong> {selectedDa
                                <p><strong>Tasa interes: </strong>{s
.interes} %</p>
                                <p><strong>Cuota UF:</strong> {selec
.precio_cuota}</p>
                                <p><strong>Cuota CLP:</strong> {clpV
                                <p><strong>Cuota UTM:</strong> {utmV
                                <p><strong>Cuota Euro:</strong> {eur
                                <p><strong>Cuota Dolar:</strong> {do
                            </div>
                        </div>
                        <button className="btn btn-primary" onClick=
>Cerrar</button>
                    </div>
                ) : (
                    <div>Cargando...</div>
                )}
            </Modal>
        </div>
    );
}
```

Terms Pricing Privacy Cookie Policy Security Community Documentation