

https://sonarcloud.io/project/issues?impactSeverities=MEDIUM&impactSoftwareQualities=MAINTAINABILITY&resolved=false&id=Tomaxido_INF2... 1/5



```

54         try {
55             const response = await axios.get(ufUrl);
56             if (response.data && response.data.UFs && response.data.
    ) {
57                 const ufValue = parseFloat(response.data.UFs[0].Valor
58                 return monto * ufValue;
59             } else {
60                 console.error(
61                 "No se encontró el valor del UF para la fecha especificada.");
62                 return null;
63             }
64         } catch (error) {
65             console.error("Error fetching UF data:", error);
66             return null;
67         }
68     };
69
70     const getUFValueByDate2 = async (date, monto) => {
71         const [year, month, day] = date.split("-");
72         let ufUrl = `https://api.cmfchile.cl/api-sbifv3/recursos_api
73         month}/dias/${day}?apikey=931fadd3fa3041a89f09ff5dc4712fc66729df50&f
74
75         try {
76             const response = await axios.get(ufUrl);
77             if (response.data && response.data.UFs && response.data.
78         ) {
79                 const ufValue = parseFloat(response.data.UFs[0].Valor
80                 return monto / ufValue;
81             } else {
82                 console.error(
83                 "No se encontró el valor del UF para la fecha especificada.");
84                 return null;
85             }
86         } catch (error) {
87             console.error("Error fetching UF data:", error);
88             return null;
89         }
90     };
91
92     const getUTMValueByDate = async (date, monto) => {
93         const [year, month, day] = date.split("-");
94         let utmUrl = `https://api.cmfchile.cl/api-sbifv3/recursos_ap
95         }/${month}?apikey=931fadd3fa3041a89f09ff5dc4712fc66729df50&formato=
96
97         try {
98             const response = await axios.get(utmUrl);
99             if (response.data && response.data.UTMs && response
100             .data.UTMs.length > 0) {
101                 const utmValue = parseFloat(response.data.UTMs[0].Va
102                 return monto / utmValue;
103             } else {
104                 console.error(
105                 "No se encontró el valor del UF para la fecha especificada.");
106                 return null;
107             }
108         } catch (error) {
109             console.error("Error fetching UF data:", error);
110             return null;
111         }
112     };
113
114     const getDolarValueByDate = async (date, monto) => {
115         const [year, month, day] = date.split("-");
116         let dolarUrl = `https://api.cmfchile.cl/api-sbifv3/recursos_
117         year}/${month}/dias/${day
118         }?apikey=931fadd3fa3041a89f09ff5dc4712fc66729df50&formato=json`;
119
120         try {
121             const response = await axios.get(dolarUrl);
122             if (response.data && response.data.Dolares && response
123             .data.Dolares.length > 0) {
124                 const dolarValue = parseFloat(response.data.Dolares[
125                 ].Valor.replace(',', '.'));
126                 return monto / dolarValue;
127             } else {
128                 console.error(
129                 "No se encontró el valor del UF para la fecha especificada.");
130                 return null;
131             }
132         }
133     };
134 }
135
136 export default {
137     getUFValueByDate,
138     getUFValueByDate2,
139     getUTMValueByDate,
140     getDolarValueByDate
141 };
142
143 // Ejemplo de uso:
144 // const result = getUFValueByDate("2023-01-01", 1000);
145 // console.log(result); // 1000
146
147 // const result2 = getUFValueByDate2("2023-01-01", 1000);
148 // console.log(result2); // 1000
149
150 // const result3 = getUTMValueByDate("2023-01-01", 1000);
151 // console.log(result3); // 1000
152
153 // const result4 = getDolarValueByDate("2023-01-01", 1000);
154 // console.log(result4); // 1000
155
156 // const result5 = getDolarValueByDate2("2023-01-01", 1000);
157 // console.log(result5); // 1000
158
159 // const result6 = getDolarValueByDate3("2023-01-01", 1000);
160 // console.log(result6); // 1000
161
162 // const result7 = getDolarValueByDate4("2023-01-01", 1000);
163 // console.log(result7); // 1000
164
165 // const result8 = getDolarValueByDate5("2023-01-01", 1000);
166 // console.log(result8); // 1000
167
168 // const result9 = getDolarValueByDate6("2023-01-01", 1000);
169 // console.log(result9); // 1000
170
171 // const result10 = getDolarValueByDate7("2023-01-01", 1000);
172 // console.log(result10); // 1000
173
174 // const result11 = getDolarValueByDate8("2023-01-01", 1000);
175 // console.log(result11); // 1000
176
177 // const result12 = getDolarValueByDate9("2023-01-01", 1000);
178 // console.log(result12); // 1000
179
180 // const result13 = getDolarValueByDate10("2023-01-01", 1000);
181 // console.log(result13); // 1000
182
183 // const result14 = getDolarValueByDate11("2023-01-01", 1000);
184 // console.log(result14); // 1000
185
186 // const result15 = getDolarValueByDate12("2023-01-01", 1000);
187 // console.log(result15); // 1000
188
189 // const result16 = getDolarValueByDate13("2023-01-01", 1000);
190 // console.log(result16); // 1000
191
192 // const result17 = getDolarValueByDate14("2023-01-01", 1000);
193 // console.log(result17); // 1000
194
195 // const result18 = getDolarValueByDate15("2023-01-01", 1000);
196 // console.log(result18); // 1000
197
198 // const result19 = getDolarValueByDate16("2023-01-01", 1000);
199 // console.log(result19); // 1000
200
201 // const result20 = getDolarValueByDate17("2023-01-01", 1000);
202 // console.log(result20); // 1000
203
204 // const result21 = getDolarValueByDate18("2023-01-01", 1000);
205 // console.log(result21); // 1000
206
207 // const result22 = getDolarValueByDate19("2023-01-01", 1000);
208 // console.log(result22); // 1000
209
210 // const result23 = getDolarValueByDate20("2023-01-01", 1000);
211 // console.log(result23); // 1000
212
213 // const result24 = getDolarValueByDate21("2023-01-01", 1000);
214 // console.log(result24); // 1000
215
216 // const result25 = getDolarValueByDate22("2023-01-01", 1000);
217 // console.log(result25); // 1000
218
219 // const result26 = getDolarValueByDate23("2023-01-01", 1000);
220 // console.log(result26); // 1000
221
222 // const result27 = getDolarValueByDate24("2023-01-01", 1000);
223 // console.log(result27); // 1000
224
225 // const result28 = getDolarValueByDate25("2023-01-01", 1000);
226 // console.log(result28); // 1000
227
228 // const result29 = getDolarValueByDate26("2023-01-01", 1000);
229 // console.log(result29); // 1000
230
231 // const result30 = getDolarValueByDate27("2023-01-01", 1000);
232 // console.log(result30); // 1000
233
234 // const result31 = getDolarValueByDate28("2023-01-01", 1000);
235 // console.log(result31); // 1000
236
237 // const result32 = getDolarValueByDate29("2023-01-01", 1000);
238 // console.log(result32); // 1000
239
240 // const result33 = getDolarValueByDate30("2023-01-01", 1000);
241 // console.log(result33); // 1000
242
243 // const result34 = getDolarValueByDate31("2023-01-01", 1000);
244 // console.log(result34); // 1000
245
246 // const result35 = getDolarValueByDate32("2023-01-01", 1000);
247 // console.log(result35); // 1000
248
249 // const result36 = getDolarValueByDate33("2023-01-01", 1000);
250 // console.log(result36); // 1000
251
252 // const result37 = getDolarValueByDate34("2023-01-01", 1000);
253 // console.log(result37); // 1000
254
255 // const result38 = getDolarValueByDate35("2023-01-01", 1000);
256 // console.log(result38); // 1000
257
258 // const result39 = getDolarValueByDate36("2023-01-01", 1000);
259 // console.log(result39); // 1000
260
261 // const result40 = getDolarValueByDate37("2023-01-01", 1000);
262 // console.log(result40); // 1000
263
264 // const result41 = getDolarValueByDate38("2023-01-01", 1000);
265 // console.log(result41); // 1000
266
267 // const result42 = getDolarValueByDate39("2023-01-01", 1000);
268 // console.log(result42); // 1000
269
270 // const result43 = getDolarValueByDate40("2023-01-01", 1000);
271 // console.log(result43); // 1000
272
273 // const result44 = getDolarValueByDate41("2023-01-01", 1000);
274 // console.log(result44); // 1000
275
276 // const result45 = getDolarValueByDate42("2023-01-01", 1000);
277 // console.log(result45); // 1000
278
279 // const result46 = getDolarValueByDate43("2023-01-01", 1000);
280 // console.log(result46); // 1000
281
282 // const result47 = getDolarValueByDate44("2023-01-01", 1000);
283 // console.log(result47); // 1000
284
285 // const result48 = getDolarValueByDate45("2023-01-01", 1000);
286 // console.log(result48); // 1000
287
288 // const result49 = getDolarValueByDate46("2023-01-01", 1000);
289 // console.log(result49); // 1000
290
291 // const result50 = getDolarValueByDate47("2023-01-01", 1000);
292 // console.log(result50); // 1000
293
294 // const result51 = getDolarValueByDate48("2023-01-01", 1000);
295 // console.log(result51); // 1000
296
297 // const result52 = getDolarValueByDate49("2023-01-01", 1000);
298 // console.log(result52); // 1000
299
300 // const result53 = getDolarValueByDate50("2023-01-01", 100
```



```

126     };
127
128     useEffect(() => {
129       if (cargoCookie !== 2) {
130         window.location.href = '/';
131       }
132     }, []);
133
134     const { data, error } = useSWR("/solicitud/all", {
135       fetcher: getAllSolis,
136       initialData: [],
137       revalidateOnMount: true,
138     });
139
140     useEffect(() => {
141       if (data) {
142         setState((prevState) => ({ ...prevState, informData: da
143       }
144     }, [data]);
145
146     const openModal = async (item) => {
147       setSelectedData(item);
148       setModalIsOpen(true);
149
150       // Fetch the conversion values when the modal is opened
151       const monto = Number(item.precio_cuota.replace(',', '.'));
152       const clpValueC = await getUFValueByDate(item.fecha, monto);
153
154       const euroValueC = await getEuroValueByDate(item.fecha, clpV
155       const utmValueC = await getUTMValueByDate(item.fecha, clpVal
156       const dolarValueC = await getDolarValueByDate(item.fecha, cl
157
158       const clpValueM = item.monto_total;
159       const euroValueM = await getEuroValueByDate(item.fecha, clpV
160       const ufValueM = await getUFValueByDate2(item.fecha, clpValu
161       const utmValueM = await getUTMValueByDate(item.fecha, clpVal
162       const dolarValueM = await getDolarValueByDate(item.fecha, cl
163
164       setEuroValueC(euroValueC.toFixed(2));
165       setUtmValueC(utmValueC.toFixed(2));
166       setCLPValueC(clpValueC.toFixed(2));
167       setDolarValueC(dolarValueC.toFixed(2));
168
169       setEuroValueM(euroValueM.toFixed(2));
170       setUtmValueM(utmValueM.toFixed(2));
171       setUFValueM(ufValueM.toFixed(2));
172       setDolarValueM(dolarValueM.toFixed(2))
173     };
174
175     const closeModal = () => {
176       setModalIsOpen(false);
177       setSelectedData(null);
178
179       setEuroValueC(null);
180       setUtmValueC(null);
181       setCLPValueC(null);
182       setDolarValueC(null);
183
184       setEuroValueM(null);
185       setUtmValueM(null);
186       setUFValueM(null);
187       setDolarValueM(null)
188     };
189
190     if (error) return <div>Error loading data</div>;
191     if (!data) return <div>Loading...</div>;
192
193     return (
194       <div className="container mt-4 border border-primary" style=
195         '100%', margin: '0 auto' >>
196         <div>
197           <h2>Solicitudes a convertir</h2>
198           <table>
199             <thead>
200               <tr>
201                 <th>Rut</th>
202                 <th>Nombre</th>
203

```



```

212         <td>{item.rut}</td>
213         <td>{item.nombre}</td>
214         <td>{item.fecha}</td>
215         <td>
216             <button className="btn btn-s
onClick=
{() => openModal(item)}>Ver</button>
217         </td>
218     </tr>
219     );
220     }}
221 </tbody>
222 </table>
223 </div>
224
225 <Modal
226     isOpen={modalIsOpen}
227     onRequestClose={closeModal}
228     contentLabel="Detalles de la Solicitud"
229     style={{
230         overlay: {
231             backgroundColor: 'rgba(0, 0, 0, 0.75)',
232         },
233         content: {
234             color: 'black',
235             top: '50%',
236             left: '50%',
237             right: '800%',
238             bottom: 'auto',
239             marginRight: '-50%',
240             transform: 'translate(-50%, -50%)',
241         },
242     }}
243 >
244     {selectedData ? (
245         <div>
246             <h2>Conversiones</h2>
247             <div style={{ display: 'flex' }}>
248                 <div style={{ flex: 1, paddingRight: '10
249                     <p><strong>Nombre:</strong> {selecte
250
251                     <p><strong>Fecha:</strong> {selected
252
253                     <p><strong>Monto UF:</strong> {ufVal
254                     <p><strong>Monto CLP:</strong> {sele
255                     .monto_total}</p>
256
257                     <p><strong>Monto UTM:</strong> {utmV
258                     <p><strong>Monto Euro:</strong> {eur
259                     <p><strong>Monto Dolar:</strong> {do
260
261                     </div>
262                     <div style={{ flex: 1, paddingLeft: '10p
263                         <p><strong>RUT:</strong> {selectedDa
264                         <p><strong>Tasa interes: </strong>{s
265                     .interes} %</p>
266
267                     <p><strong>Cuota UF:</strong> {selec
268                     .precio_cuota}</p>
269
270                     <p><strong>Cuota CLP:</strong> {clpV
271                     <p><strong>Cuota UTM:</strong> {utmV
272                     <p><strong>Cuota Euro:</strong> {eur
273                     <p><strong>Cuota Dolar:</strong> {do
274
275                     </div>
276                     </div>
277                     <button className="btn btn-primary" onClick=
278                     >Cerrar</button>
279                 </div>
280             ) : (
281                 <div>Cargando...</div>
282             )}
283         </Modal>
284     </div>
285 );
286 }

```

