GESTOR DE PRECIOS - ICG / ÁGORA / STOCKAGILE (Versión Final Optimizada)

==

IMPORTANTE: PARA PODER EJECUTAR ESTE PROGRAMA HAY QUE INSTALAR NODE.JS Y SUS RESPECTIVAS LIBRERIAS EN VUESTRO ORDENADOR

Cómo instalar y ejecutar:

1. Crea una carpeta nueva en tu equipo y mete estos 5 archivos.
    - Mantén la estructura: `public/index.html` y `src/App.jsx`.
2. Necesitas tener instalado **Node.js** en tu ordenador.
3. Abre una terminal CMD en esa carpeta y ejecuta: npm install
   4. Para probarlo: npm start
   5. Para crear el ejecutable final (.exe): npm run dist
   (Esto creará una carpeta `dist` con el instalador dentro).

1. Configuración (package.json)

Define las dependencias y los comandos de arranque.

```
{
  "name": "gestor-precios-icg-agora",
  "version": "1.0.0",
  "description": "Gestor de Exportación SQL Optimizado",
  "main": "main.js",
  "scripts": {
    "start": "webpack --mode development && electron .",
    "dist": "webpack --mode production && electron-builder"
  },
  "author": "Tu Nombre",
  "license": "ISC",
  "dependencies": {
    "react": "^18.2.0",
    "react-dom": "^18.2.0",
    "mssql": "^10.0.1",
    "xlsx": "^0.18.5",
    "lucide-react": "^0.263.1"
  },
  "devDependencies": {
    "electron": "^28.1.0",
    "electron-builder": "^24.9.1",
    "@babel/core": "^7.23.0",
    "@babel/preset-env": "^7.23.0",
    "@babel/preset-react": "^7.23.0",
    "babel-loader": "^9.1.3",
    "css-loader": "^6.8.1",
    "style-loader": "^3.3.3",
    "webpack": "^5.89.0",
    "webpack-cli": "^5.1.4",
    "postcss": "^8.4.31",
    "postcss-loader": "^7.3.3",
    "tailwindcss": "^3.3.3",
    "autoprefixer": "^10.4.16"
  }
}
```

2. Compilador (webpack.config.js)

Traduce el código React para que Electron lo entienda.

```
const path = require('path');

module.exports = {
  mode: 'development',
  entry: './src/index.js',
  target: 'electron-renderer',
  devtool: 'source-map',
  module: {
    rules: [
      {
        test: /\.(js|jsx)$/,
        exclude: /node_modules/,
        use: {
          loader: 'babel-loader',
          options: {
            presets: ['@babel/preset-env', '@babel/preset-react']
          }
        }
      },
      {
        test: /\.css$/i,
        use: ['style-loader', 'css-loader', 'postcss-loader'],
      },
    ],
  },
  resolve: {
    extensions: ['.js', '.jsx'],
  },
  output: {
    filename: 'bundle.js',
    path: path.resolve(__dirname, 'public'),
  },
};
```

3. Backend Electron (main.js)

Maneja la conexión SQL persistente, timeouts largos y procesos en segundo plano.

```
const { app, BrowserWindow, ipcMain } = require('electron');
const path = require('path');
const sql = require('mssql');

let mainWindow;
let dbPool = null;

function createWindow() {
  mainWindow = new BrowserWindow({
    width: 1280,
    height: 800,
    title: "Gestor de Precios - Exportación ICG",
    webPreferences: {
      preload: path.join(__dirname, 'preload.js'),
      nodeIntegration: false,
      contextIsolation: true
    }
  });

  mainWindow.loadFile('public/index.html');
}
```

```
app.whenReady().then(createWindow);

app.on('window-all-closed', () => {
  if (process.platform !== 'darwin') app.quit();
});

// --- MANEJADORES DE BASE DE DATOS (IPC) ---

ipcMain.handle('db-connect', async (event, config) => {
  try {
    if (dbPool) {
      await dbPool.close();
      dbPool = null;
    }

    const sqlConfig = {
      user: config.user,
      password: config.password,
      database: config.database,
      server: config.server,
      pool: {
        max: 20,
        min: 0,
        idleTimeoutMillis: 30000
      },
      options: {
        encrypt: false,
        trustServerCertificate: true,
        enableArithAbort: true,
        requestTimeout: 300000 // 5 minutos de timeout
      }
    };

    dbPool = await new sql.ConnectionPool(sqlConfig).connect();
    console.log("Conexión SQL establecida con éxito.");
    return { success: true, message: "Conectado correctamente" };
  } catch (err) {
    console.error("Error SQL Connect:", err);
    dbPool = null;
    return { success: false, message: err.message };
  }
});
```

```
ipcMain.handle('db-execute', async (event, query) => {
  try {
    if (!dbPool || !dbPool.connected) {
        throw new Error("No hay conexión activa. Por favor, conéctese de nuevo.");
    }

    const request = dbPool.request();
    request.timeout = 300000; // 5 minutos

    const result = await request.query(query);

    return { success: true, data: result.recordset, rowsAffected: result.rowsAffected };
  } catch (err) {
    console.error("Error SQL Query:", err);
    return { success: false, message: err.message };
  }
});
```

4. Puente de Seguridad (preload.js)

Conecta la interfaz con el backend de forma segura.

```
const { contextBridge, ipcRenderer } = require('electron');

contextBridge.exposeInMainWorld('electronAPI', {
  connectDB: (config) => ipcRenderer.invoke('db-connect', config),
  executeSQL: (query) => ipcRenderer.invoke('db-execute', query),
  log: (msg) => console.log(msg)
});
```

5. Web Entry (public/index.html)

El HTML base que carga los estilos y el script compilado.

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <meta http-equiv="Content-Security-Policy" content="default-src 'self' 'unsafe-inline'
[https://cdn.tailwindcss.com](https://cdn.tailwindcss.com)
```

```
[https://cdnjs.cloudflare.com](https://cdnjs.cloudflare.com) data:;">
    <title>Gestor de Precios - Exportación ICG</title>
    <script src="[https://cdn.tailwindcss.com](https://cdn.tailwindcss.com)"></script>
  </head>
  <body class="bg-slate-50">
    <div id="root"></div>
    <script src="bundle.js"></script>
  </body>
</html>
```

6. React Entry (src/index.js)

Inicializa la aplicación React.

```
import React from 'react';
import { createRoot } from 'react-dom/client';
import App from './App';

const container = document.getElementById('root');
const root = createRoot(container);
root.render(<App />);
```

7. Aplicación Principal (src/App.jsx)

La lógica completa: Plantillas SQL, búsqueda de tarifas, optimización de tabla y exportación Excel.

```
import React, { useState, useEffect } from 'react';
import { Database, Code, Copy, CheckCircle, AlertCircle, Server, Lock, User, LogOut, Play,
Table as TableIcon, FileSpreadsheet, Shirt, UtensilsCrossed, Search, Loader2 } from
'lucide-react';

// --- PLANTILLA SQL ÁGORA (COMPLETA) ---
const AGORA_SQL_TEMPLATE = `/*
CREATE FUNCTION udf_decimal_a_base(@N bigint, @Base int)
RETURNS varchar(200)
AS
BEGIN
-- ... (Función asumida existente en BBDD) ...
END
GO
```

```
*/

SELECT DISTINCT
REPLACE(replace(sec.DESCRIPCION,'SECCION ',''),'SECCIÓN ','') as 'Familia',
CASE
	WHEN favo.Categorias IS NULL THEN ''
	ELSE favo.Categorias
END as 'Categorías',
'' as 'Alérgenos',
'' as 'Etiquetas',
art.DESCRIPCION as 'Producto',
case
	when fo.DESCRIPCION is null then ''
	else fo.DESCRIPCION+' '+art.DESCRIPCION
end as 'Formato',
case
	when pv1.CODFORMATO=0 then ''
	else replace(cast(form.DOSIS1 as varchar),'.',',')
end as 'Ratio',
'' as 'Código Barras',
'' as 'PLU',
case
	when pv1.CODFORMATO=0 then replace(CAST(art.ultimocoste AS VARCHAR),'.',',')
	else REPLACE('ASISMAN','ASISMAN','')
end as 'Precio Coste',
case
	when pv1.CODFORMATO=0 then CAST(tta.PORCENTAJE AS VARCHAR)
	else REPLACE('ASISMAN','ASISMAN','')
end as '% Impuesto',
case
	when pv1.CODFORMATO=0 then CAST(ttacomp.PORCENTAJE AS VARCHAR)
	else REPLACE('ASISMAN','ASISMAN','')
end as '% Impuesto Compra',
case
	when fo.DESCRIPCION is null THEN 'Unidad'
	else ''
end as 'Unidad de Compra',
case
	when art.USASTOCKS='T' and pv1.CODFORMATO=0 THEN 'Sí'
	else 'No'
end as 'Control Stock',
case
	when fo.DESCRIPCION is null THEN 'Unidad'
```

```
        else ''
end as 'Unidad de Medida',
'Sí' as 'Vendible Principal',
'No' as 'Vendible Añadido',
case
        when art.PORPESO='F' THEN 'No'
        else 'Sí'
end as 'Venta Peso',
case
        when art.COLORFONDO>0 then
'#'+left('000000',6-len([dbo].[udf_decimal_a_base](art.colorfondo,16)))+[dbo].[udf_decimal_a_base](art.colorfondo,16)
        else ''
end as 'Color',
case
        when fo.DESCRIPCION is null then art.DESCRIPCION
        else fo.DESCRIPCION+' '+art.DESCRIPCION
end as 'Texto Botón',
'' as 'Imagen',
case
        when art.ORDEN!=0 AND pv1.CODFORMATO=0 and situart.impcocinaart is not null
then situart.impcocinaart
        when art.ORDEN!=0 AND pv1.CODFORMATO=0 and situart.impcocinaart is null and
situ.impcocina is not null then situ.impcocina
        when art.ORDEN!=0 AND pv1.CODFORMATO=0 and situart.impcocinaart is null and
situ.impcocina is null then 'SIN SITUACION'
else '' end as 'Tipo de Preparación',
case
        when pv1.CODFORMATO=0 and art.ORDEN=1 then 'PRIMEROS'
        when pv1.CODFORMATO=0 and art.ORDEN=2 then 'SEGUNDOS'
        when pv1.CODFORMATO=0 and art.ORDEN=3 then 'TERCEROS'
        when pv1.CODFORMATO=0 and art.ORDEN=4 then 'CUARTOS'
        when pv1.CODFORMATO=0 and art.ORDEN=5 then 'QUINTOS'
        when pv1.CODFORMATO=0 and art.ORDEN=6 then 'SEXTOS'
        when pv1.CODFORMATO=0 and art.ORDEN=7 then 'SEPTIMOS'
        when pv1.CODFORMATO=0 and art.ORDEN=8 then 'OCTAVOS'
        when pv1.CODFORMATO=0 and art.ORDEN=9 then 'NOVENOS'
        when pv1.CODFORMATO=0 and art.ORDEN=10 then 'DECIMOS'
        WHEN pv1.CODFORMATO>0 then ''
else '' end as 'Orden de Preparación',
case
        when fo.DESCRIPCION is null then art.DESCRIPCION
        else fo.DESCRIPCION+' '+art.DESCRIPCION
```

```
end as 'Texto Documento',
case
	when fo.DESCRIPCION is null then art.DESCRIPCION
	else fo.DESCRIPCION+' '+art.DESCRIPCION
end as 'Texto Comanda',
'' as 'Texto Ficha',
'' as 'Imagen Ficha',
case
	when (pv1.DESDE2 <= GETDATE() AND PV1.HASTA2 >= GETDATE()) AND PV1.VALOR2 is null AND PV1.VALOR IS NULL then 0
	when (pv1.DESDE2 <= GETDATE() AND PV1.HASTA2 >= GETDATE()) AND PV1.VALOR2 is null AND PV1.VALOR IS NOT NULL then PV1.VALOR
	when (pv1.DESDE2 <= GETDATE() AND PV1.HASTA2 >= GETDATE()) AND PV1.VALOR2 is NOT null then PV1.VALOR2
	when (pv1.DESDE2 > GETDATE() OR PV1.HASTA2 < GETDATE()) AND PV1.VALOR is null then 0
	else PV1.VALOR
end as 'PP Barra',
0 as 'PA Barra',
0 as 'PM Barra',
case
	when (pv2.DESDE2 <= GETDATE() AND PV2.HASTA2 >= GETDATE()) AND PV2.VALOR2 is null AND PV2.VALOR IS NULL then 0
	when (pv2.DESDE2 <= GETDATE() AND PV2.HASTA2 >= GETDATE()) AND PV2.VALOR2 is null AND PV2.VALOR IS NOT NULL then PV2.VALOR
	when (pv2.DESDE2 <= GETDATE() AND PV2.HASTA2 >= GETDATE()) AND PV2.VALOR2 is NOT null then PV2.VALOR2
	when (pv2.DESDE2 > GETDATE() OR PV2.HASTA2 < GETDATE()) AND PV2.VALOR is null then 0
	else PV2.VALOR
end as 'PP Terraza',
0 as 'PA Terraza',
0 as 'PP Terraza',
case
	when (pv3.DESDE2 <= GETDATE() AND PV3.HASTA2 >= GETDATE()) AND PV3.VALOR2 is null AND PV3.VALOR IS NULL then 0
	when (pv3.DESDE2 <= GETDATE() AND PV3.HASTA2 >= GETDATE()) AND PV3.VALOR2 is null AND PV3.VALOR IS NOT NULL then PV3.VALOR
	when (pv3.DESDE2 <= GETDATE() AND PV3.HASTA2 >= GETDATE()) AND PV3.VALOR2 is NOT null then PV3.VALOR2
	when (pv3.DESDE2 > GETDATE() OR PV3.HASTA2 < GETDATE()) AND PV3.VALOR is null then 0
	else pv3.VALOR
```

```
end as 'PP Salón',
0 as 'PA Salón',
0 as 'PM Salón'

FROM PRECIOSVENTA pv1
left join ARTICULOS ART on pv1.CODARTICULO=art.CODARTICULO
left join formatos fo on pv1.CODFORMATO=fo.CODFORMATO
left join SECCIONES sec on art.DPTO=sec.DPTO and art.SECCION=sec.SECCION
left join(SELECT sitf.[CODSECCION],STRING_AGG(sit.descripcion,',' )as impcocina
                    FROM SITUACIONESFAMILIA sitf LEFT JOIN situaciones sit ON
sitf.CODSITUACION=sit.CODSITUACION
                    GROUP BY sitf.CODSECCION) situ on art.SECCION=situ.CODSECCION
left join(SELECT  [CODARTICULO],STRING_AGG(sit.DESCRIPCION,',')as impcocinaart
                    FROM SITUACIONESARTICULO sitf LEFT JOIN situaciones sit ON
sitf.CODSITUACION=sit.CODSITUACION
                    GROUP BY sitf.CODARTICULO)situart on
art.CODARTICULO=situart.CODARTICULO
left join FORMATOS FORM on pv1.CODFORMATO=form.CODFORMATO
left join (SELECT CODARTICULO, STRING_AGG(tpf.DESCRIPCION, ',') as Categorias
                     FROM favoritos fav LEFT JOIN TIPOFAVORITOS TPF ON
FAV.CODFAVORITO=TPF.CODFAVORITO
                     where fav.CODFAVORITO in (select distinct CODFAVORITO from
FAVORITOSTIPOSTERMINAL)
                     GROUP BY CODARTICULO)favo on art.CODARTICULO=favo.CODARTICULO
left join PRECIOSVENTA pv2 on art.CODARTICULO=pv2.CODARTICULO and
pv1.CODFORMATO=pv2.CODFORMATO
left join PRECIOSVENTA pv3 on art.CODARTICULO=pv3.CODARTICULO and
pv1.CODFORMATO=pv3.CODFORMATO
left join TASAS ta on art.CODTASA1=ta.CODTASA
left join TIPOTASAS tta on ta.CODTIPOTASA1=tta.CODTIPOTASA
left join TASAS tacomp on art.CODTASA1C=tacomp.CODTASA
left join TIPOTASAS ttacomp on tacomp.CODTIPOTASA1=ttacomp.CODTIPOTASA

WHERE art.DESCATALOGADO='F' AND
(
((pv1.IDTARIFAV=1 and PV1.DESCATALOGADO=0) AND (pv2.IDTARIFAV=2 and
PV2.DESCATALOGADO=0) AND (pv3.IDTARIFAV=4 and PV3.DESCATALOGADO=0)) OR
(((pv1.IDTARIFAV=1 and PV1.DESCATALOGADO=0) AND (pv2.IDTARIFAV=1 and
PV2.DESCATALOGADO=0) AND (pv3.IDTARIFAV=1 and PV3.DESCATALOGADO=0)) AND
(PV2.VALOR = 0 OR PV3.VALOR = 0))
)

UNION
```

```
SELECT distinct
REPLACE(replace(sec.DESCRIPCION,'SECCION ',''),'SECCIÓN ','') as 'Familia',
CASE
        WHEN favo.Categorias IS NULL THEN ''
        ELSE favo.Categorias
END as 'Categorías',
'' as 'Alérgenos',
'' as 'Etiquetas',
art.DESCRIPCION as 'Producto',
 '' as 'Formato',
'' as 'Ratio',
'' as 'Código Barras',
'' as 'PLU',
CAST(art.ultimocoste AS VARCHAR) as 'Precio Coste',
CAST(TTA.PORCENTAJE AS VARCHAR) as '% Impuesto',
CAST(ttacomp.PORCENTAJE AS VARCHAR) as '% Impuesto Compra',
'Unidad' as 'Unidad de Compra',
case
        when art.USASTOCKS='T' THEN 'Sí'
        else 'No'
end as 'Control Stock',
'Unidad' as 'Unidad de Medida',
'No' as 'Vendible Principal',
'No' as 'Vendible Añadido',
'No'as 'Venta Peso',
case
        when art.COLORFONDO>0 then
'#'+left('000000',6-len([dbo].[udf_decimal_a_base](art.colorfondo,16)))+[dbo].[udf_decimal_
a_base](art.colorfondo,16)
        else ''
end as 'Color',
art.DESCRIPCION as 'Texto Botón',
'' as 'Imagen',
case
        when art.ORDEN!=0 AND pv1.CODFORMATO=0 and situart.impcocinaart is not null
then situart.impcocinaart
        when art.ORDEN!=0 AND pv1.CODFORMATO=0 and situart.impcocinaart is null and
situ.impcocina is not null then situ.impcocina
        when art.ORDEN!=0 AND pv1.CODFORMATO=0 and situart.impcocinaart is null and
situ.impcocina is null then 'SIN SITUACION'
        else '' end as 'Tipo de Preparación',
case
```

```
        when pv1.CODFORMATO=0 and art.ORDEN=1 then 'PRIMEROS'
        when pv1.CODFORMATO=0 and art.ORDEN=2 then 'SEGUNDOS'
        when pv1.CODFORMATO=0 and art.ORDEN=3 then 'TERCEROS'
        when pv1.CODFORMATO=0 and art.ORDEN=4 then 'CUARTOS'
        when pv1.CODFORMATO=0 and art.ORDEN=5 then 'QUINTOS'
        when pv1.CODFORMATO=0 and art.ORDEN=6 then 'SEXTOS'
        when pv1.CODFORMATO=0 and art.ORDEN=7 then 'SEPTIMOS'
        when pv1.CODFORMATO=0 and art.ORDEN=8 then 'OCTAVOS'
        when pv1.CODFORMATO=0 and art.ORDEN=9 then 'NOVENOS'
        WHEN pv1.CODFORMATO>0 then ''
else '' end as 'Orden de Preparación',
art.DESCRIPCION  as 'Texto Documento',
art.DESCRIPCION  as 'Texto Comanda',
'' as 'Texto Ficha',
'' as 'Imagen Ficha',
0 as 'PP Barra',
0 as 'PA Barra',
0 as 'PM Barra',
0 as 'PP Terraza',
0 as 'PA Terraza',
0 as 'PP Terraza',
0 as 'PP Salón',
0 as 'PA Salón',
0 as 'PM Salón'

FROM PRECIOSVENTA pv1
left join ARTICULOS ART on pv1.CODARTICULO=art.CODARTICULO
left join formatos fo on pv1.CODFORMATO=fo.CODFORMATO
left join FORMATOS FORM on pv1.CODFORMATO=form.CODFORMATO
left join SECCIONES sec on art.DPTO=sec.DPTO and art.SECCION=sec.SECCION
left join(SELECT sitf.[CODSECCION],STRING_AGG(sit.descripcion,',' )as impcocina
                 FROM SITUACIONESFAMILIA sitf LEFT JOIN situaciones sit ON
sitf.CODSITUACION=sit.CODSITUACION
                 GROUP BY sitf.CODSECCION)situ on art.SECCION=situ.CODSECCION
left join(SELECT sitf.[CODARTICULO],STRING_AGG(sit.DESCRIPCION,',')as impcocinaart
                 FROM SITUACIONESARTICULO sitf LEFT JOIN situaciones sit ON
sitf.CODSITUACION=sit.CODSITUACION
                 GROUP BY sitf.CODARTICULO)lsituart on
art.CODARTICULO=situart.CODARTICULO
left join(SELECT CODARTICULO, STRING_AGG(tpf.DESCRIPCION, ',') as Categorias
                 FROM favoritos fav LEFT JOIN TIPOFAVORITOS TPF ON
FAV.CODFAVORITO=TPF.CODFAVORITO
                 where fav.CODFAVORITO in (select distinct CODFAVORITO from
FAVORITOSTIPOSTERMINAL)
                    GROUP BY CODARTICULO)favo on art.CODARTICULO=favo.CODARTICULO
left join PRECIOSVENTA pv2 on art.CODARTICULO=pv2.CODARTICULO and
pv1.CODFORMATO=pv2.CODFORMATO
left join PRECIOSVENTA pv3 on art.CODARTICULO=pv3.CODARTICULO and
pv1.CODFORMATO=pv3.CODFORMATO
left join TASAS ta on art.CODTASA1=ta.CODTASA
left join TIPOTASAS tta on ta.CODTIPOTASA1=tta.CODTIPOTASA
left join TASAS tacomp on art.CODTASA1C=tacomp.CODTASA
left join TIPOTASAS ttacomp on tacomp.CODTIPOTASA1=ttacomp.CODTIPOTASA

WHERE
(
((pv1.IDTARIFAV=1 and PV1.DESCATALOGADO=0) AND (pv2.IDTARIFAV=2 and
PV2.DESCATALOGADO=0) AND (pv3.IDTARIFAV=4 and PV3.DESCATALOGADO=0)) OR
(((pv1.IDTARIFAV=1 and PV1.DESCATALOGADO=0) AND (pv2.IDTARIFAV=1 and
PV2.DESCATALOGADO=0) AND (pv3.IDTARIFAV=1 and PV3.DESCATALOGADO=0)) AND
(PV2.VALOR = 0 OR PV3.VALOR = 0))
)
          and art.DESCATALOGADO='F'
          and art.CODARTICULO in (SELECT distinct art.CODARTICULO as Producto
                                                                      FROM PRECIOSVENTA pv1 left join
ARTICULOS ART on pv1.CODARTICULO=art.CODARTICULO
                                                                                left join formatos fo on
pv1.CODFORMATO=fo.CODFORMATO
                                                                                left join SECCIONES sec on
art.DPTO=sec.DPTO and art.SECCION=sec.SECCION
                                                                                left join (        SELECT
CODARTICULO, STRING_AGG(tpf.DESCRIPCION, ',') as Categorias
                                                                                                        FROM
favoritos fav LEFT JOIN TIPOFAVORITOS TPF ON FAV.CODFAVORITO=TPF.CODFAVORITO
                                                                                                        where
fav.CODFAVORITO in (select distinct CODFAVORITO from FAVORITOSTIPOSTERMINAL)
                                                                                                        GROUP BY
CODARTICULO)favo on art.CODARTICULO=favo.CODARTICULO
                                                                                left join PRECIOSVENTA pv2 on
art.CODARTICULO=pv2.CODARTICULO and pv1.CODFORMATO=pv2.CODFORMATO
                                                                                left join PRECIOSVENTA pv3 on
art.CODARTICULO=pv3.CODARTICULO and pv1.CODFORMATO=pv3.CODFORMATO
                                                                                left join TASAS ta on
art.CODTASA1=ta.CODTASA
                                                                                left join TIPOTASAS tta on
ta.CODTIPOTASA1=tta.CODTIPOTASA
```

```
                                                        left join TASAS tacomp on
art.CODTASA1C=tacomp.CODTASA
                                                        left join TIPOTASAS ttacomp on
tacomp.CODTIPOTASA1=ttacomp.CODTIPOTASA
                                                 where ((pv1.IDTARIFAV=1 and
PV1.DESCATALOGADO=0)
                                                        or (pv2.IDTARIFAV= 2 and
PV2.DESCATALOGADO=0)
                                                        or (pv3.IDTARIFAV=3 and
PV3.DESCATALOGADO=0) )
                                                        AND art.DESCATALOGADO='F' and
pv1.CODFORMATO>0)

ORDER BY Producto`;

// --- PLANTILLA SQL STOCKAGILE (COMPLETA) ---
const STOCKAGILE_SQL_TEMPLATE = `select distinct
art.codarticulo as 'Código',
case
       when art.DESCRIPCION = '' or art.DESCRIPCION is null then '~NO_NAME'
       ELSE TRIM(art.DESCRIPCION)
end as 'Nombre',
'' as 'Categoría de talla (nombre)',
'' as 'Categoría de talla (código)',
ARTL.TALLA as 'Talla (nombre)',
'' as 'Talla (código)',
'' as 'Categoría de color (nombre)',
'' as 'Categoría de color (código)',
ARTL.COLOR as 'Color (nombre)',
'' as 'Color (código)',
CONCAT(art.CODARTICULO,'-',artl.TALLA,'-',artl.COLOR) as 'SKU',
CASE
       WHEN artl.CODBARRAS IS NULL OR ARTL.CODBARRAS = '' THEN
CONCAT('~NO_CODE_', ART.CODARTICULO, ARTL.TALLA, ARTL.COLOR)
       ELSE trim(ARTL.CODBARRAS)
END as 'Código de barras',
case
       when M.DESCRIPCION is null then ''
       else M.DESCRIPCION
end AS 'Marca (nombre)',
case
       when cast(m.CODMARCA as varchar) is null then ''
       else cast(m.CODMARCA as varchar)
```

```
end as 'Marca (código)',
case
        when T.TEMPORADA is null then ''
        else T.TEMPORADA
end as 'Temporada (nombre)',
case
        when cast(T.CODTEMPORADA as varchar) is null then ''
        else cast(T.CODTEMPORADA as varchar)
end as 'Temporada (código)',
case
        when d.DESCRIPCION is null then '~NO_DPTO'
        ELSE trim(REPLACE(replace(D.DESCRIPCION,'DEPARTAMENTO ',''),'DPTO ',''))
END AS 'Categoría (nombre)',
'' as 'Categoría (código)',
CASE
        WHEN S.DESCRIPCION IS NULL THEN '~NO_SECCION'
        ELSE trim(REPLACE(replace(s.DESCRIPCION,'SECCION ',''),'SECCIÓN ',''))
END AS 'Subcategoría (nombre)',
'' as 'Subcategoría (código)',
'' as 'Etiquetas',
'' as 'Composición',
CASE
        WHEN prov.NOMPROVEEDOR IS NULL THEN '~NO_PROV'
        ELSE prov.NOMPROVEEDOR
END as 'Proveedor',
art.REFPROVEEDOR as 'Código de origen',
'' as 'Precio DDP (producto)',
'' as 'Precio DDP (variante)',
pv.PBRUTO as 'Precio con impuestos (producto)',
pv.PNETO as 'Precio con impuestos (variante)',
case
        when pv.PBRUTO<>pv.PNETO then 'VERDADERO'
        ELSE 'FALSO'
END as '¿El precio de variante es de descuento?',
CASE
        WHEN CAST(PV.PBRUTO2 AS VARCHAR) IS NULL OR PV.PBRUTO2 = 0 THEN ''
        ELSE CAST(PV.PBRUTO2 AS VARCHAR)
END as 'Precio descuento con impuestos (producto)',
CASE
        WHEN CAST(PV.PNETO2 AS VARCHAR) IS NULL OR PV.PNETO2 = 0 THEN ''
        ELSE CAST(PV.PNETO2 AS VARCHAR)
END as 'Precio descuento con impuestos (variante)',
IMPV.IVA as 'Tipo de impuesto',
case
	when ST.STOCK is null then 0
	else ST.STOCK
end as 'Existencias',
case
	when art.DESCRIPCION = '' or art.DESCRIPCION is null then '~NO_NAME'
	ELSE TRIM(art.DESCRIPCION)
end as 'Descripción',
'' as 'Descripción larga',
'' as 'Imágenes',
'' as 'Categorías Web',
'' as 'Id externo del canal',
'' as 'Atributos dinámicos',
'' as 'Ancho',
'' as 'Altura',
'' as 'Profundidad',
'' as 'Peso',
'' as 'Unidad de peso',
'' as 'Localización'

FROM ARTICULOS ART LEFT JOIN SECCIONES s ON (ART.DPTO=s.NUMDPTO AND
ART.SECCION=s.NUMSECCION)
	LEFT JOIN DEPARTAMENTO D ON (ART.DPTO=D.NUMDPTO)
	LEFT JOIN ARTICULOSLIN ARTL ON (ART.CODARTICULO = ARTL.CODARTICULO)
	LEFT JOIN PRECIOSVENTA PV ON (ARTL.CODARTICULO=PV.CODARTICULO and
artl.TALLA=pv.TALLA and artl.COLOR=pv.COLOR)
	LEFT JOIN IMPUESTOS IMPV ON (ART.TIPOIMPUESTO=IMPV.TIPOIVA)
	LEFT JOIN IMPUESTOS IMPC ON (ART.IMPUESTOCOMPRA=IMPC.TIPOIVA)
	LEFT JOIN MARCA M ON (ART.MARCA=M.CODMARCA)
	LEFT JOIN TEMPORADAS T ON (ART.TEMPORADA=T.TEMPORADA)
	left join REFERENCIASPROV refs on (art.REFPROVEEDOR=refs.REFPROVEEDOR and
art.CODARTICULO=refs.CODARTICULO)
	left join PROVEEDORES prov on (refs.CODPROVEEDOR=prov.CODPROVEEDOR)
	LEFT JOIN STOCKS ST ON (ST.CODARTICULO = ARTL.CODARTICULO AND ST.TALLA =
ARTL.TALLA AND ST.COLOR = ARTL.COLOR)

where ART.DESCATALOGADO = 'F' and art.CODARTICULO > 0
AND PV.IDTARIFAV = 1

GROUP BY s.DESCRIPCION, D.DESCRIPCION,
art.DESCRIPCION, art.COLORFONDO, ART.CODARTICULO, art.USASTOCKS, art.PORPESO,
artl.TALLA, artl.COLOR, artl.CODBARRAS,
prov.NOMPROVEEDOR, art.REFPROVEEDOR,
m.codmarca, m.DESCRIPCION, t.CODTEMPORADA, t.TEMPORADA,
IMPV.IVA, PV.PBRUTO, PV.PNETO, PV.PBRUTO2, PV.PNETO2, ST.STOCK`;

const App = () => {
  const [isConnected, setIsConnected] = useState(false);
  const [isConnecting, setIsConnecting] = useState(false);
  const [connectError, setConnectError] = useState('');

  const [dbConfig, setDbConfig] = useState({
    server: 'localhost\\SQLEXPRESS',
    user: 'sa',
    password: '',
    database: 'ICG_DB'
  });

  const [selectedProgram, setSelectedProgram] = useState('agora');
  const [configs] = useState({
    agora: {
        name: 'Ágora (Exportación ICG)',
        isExport: true,
        icon: UtensilsCrossed,
        template: AGORA_SQL_TEMPLATE,
        desc: 'Para Hostelería: Formatos, Preparación y Platos',
        tariffQuery: 'SELECT * FROM TARIFAS'
    },
    stockagile: {
        name: 'StockAgile (Exportación ICG)',
        isExport: true,
        icon: Shirt,
        template: STOCKAGILE_SQL_TEMPLATE,
        desc: 'Para Retail: Tallas, Colores y Stock',
        tariffQuery: 'SELECT * FROM TARIFASVENTA'
    }
  });

  const [selectedTariff, setSelectedTariff] = useState(1);
  const [generatedSQL, setGeneratedSQL] = useState('');
  const [copied, setCopied] = useState(false);

  const [isSearchingTariffs, setIsSearchingTariffs] = useState(false);
  const [availableTariffs, setAvailableTariffs] = useState([]);
  const [tariffError, setTariffError] = useState('');
  const [isExecuting, setIsExecuting] = useState(false);
  const [queryResults, setQueryResults] = useState(null);
  const [executionMessage, setExecutionMessage] = useState('');

  const [xlsxReady, setXlsxReady] = useState(false);

  useEffect(() => {
    const script = document.createElement('script');
    script.src =
"[https://cdnjs.cloudflare.com/ajax/libs/xlsx/0.18.5/xlsx.full.min.js](https://cdnjs.cloudflare.com/
ajax/libs/xlsx/0.18.5/xlsx.full.min.js)";
    script.async = true;
    script.onload = () => setXlsxReady(true);
    document.body.appendChild(script);

    return () => {
      document.body.removeChild(script);
    }
  }, []);

  const handleConnect = async (e) => {
    e.preventDefault();
    setIsConnecting(true);
    setConnectError('');

    if (!window.electronAPI) {
      setConnectError("Error crítico: No estás ejecutando la aplicación en Electron.");
      setIsConnecting(false);
      return;
    }

    try {
      const response = await window.electronAPI.connectDB(dbConfig);
      if (response.success) {
        setIsConnected(true);
      } else {
        setConnectError('Error al conectar: ' + response.message);
      }
    } catch (err) {
      setConnectError('Error inesperado: ' + err.message);
    } finally {
      setIsConnecting(false);
    }
```

```
 };

 const handleDisconnect = () => {
  setIsConnected(false);
  setConnectError('');
  setQueryResults(null);
  setAvailableTariffs([]);
 };

 const handleSearchTariffs = async () => {
  if (!window.electronAPI) return;

  setIsSearchingTariffs(true);
  setTariffError('');
  setAvailableTariffs([]);

  const query = configs[selectedProgram].tariffQuery;

  try {
    const response = await window.electronAPI.executeSQL(query);
    if (response.success && response.data) {
      const normalized = response.data.map(row => {
        const id = row.IDTARIFAV || row.CODTARIFA || row.CODIGO || row.ID || row.id || 0;
        const nombre = row.DESCRIPCION || row.NOMBRE || row.TARIFA || row.name || 'Sin
Nombre';
        return { id, nombre };
      });

      if (normalized.length > 0) {
        setAvailableTariffs(normalized);
      } else {
        setTariffError("No se encontraron tarifas en la base de datos.");
      }
    } else {
      setTariffError(`Error: ${response.message}`);
    }
  } catch (err) {
    setTariffError(`Error de búsqueda: ${err.message}`);
  } finally {
    setIsSearchingTariffs(false);
  }
 };
```

```
  const handleExecuteSQL = async () => {
    if (!window.electronAPI) return;

    setIsExecuting(true);
    setExecutionMessage('');
    setQueryResults(null);

    try {
      const response = await window.electronAPI.executeSQL(generatedSQL);
      if (response.success) {
        setQueryResults(response.data);
        setExecutionMessage(`Consulta ejecutada correctamente. ${response.data.length} filas recuperadas.`);
      } else {
        setExecutionMessage(`Error en la consulta: ${response.message}`);
      }
    } catch (err) {
      setExecutionMessage(`Error de ejecución: ${err.message}`);
    } finally {
      setIsExecuting(false);
    }
  };

  const handleExportExcel = () => {
    if (!queryResults || queryResults.length === 0) return;

    const XLSX = window.XLSX;

    if (!XLSX) {
        alert("La librería de Excel aún no se ha cargado. Espere un segundo e intente de nuevo.");
        return;
    }

    const fileName = `Export_${selectedProgram}_Tarifa${selectedTariff}_${new Date().toISOString().slice(0,10)}.xlsx`;

    const wb = XLSX.utils.book_new();
    const ws = XLSX.utils.json_to_sheet(queryResults);

    const colWidths = Object.keys(queryResults[0]).map(key => ({ wch: Math.max(key.length, 15) }));
    ws['!cols'] = colWidths;
```

```
      XLSX.utils.book_append_sheet(wb, ws, "Datos");
      XLSX.writeFile(wb, fileName);
  };

  useEffect(() => {
    const conf = configs[selectedProgram];
    let sql = `-- Conectado a: ${dbConfig.server} (Base: ${dbConfig.database})\n`;
    sql += `-- Programa: ${conf.name}\n`;
    sql += `-- Generado: ${new Date().toLocaleString()}\n`;
    sql += `-- Tarifa seleccionada: ${selectedTariff}\n\n`;

    let template = conf.template;

    if (selectedProgram === 'agora') {
        const regex = /pv1\.IDTARIFAV\s*=\s*\d+/gi;
        template = template.replace(regex, `pv1.IDTARIFAV=${selectedTariff}`);
    } else if (selectedProgram === 'stockagile') {
        const regex = /PV\.IDTARIFAV\s*=\s*\d+/gi;
        template = template.replace(regex, `PV.IDTARIFAV = ${selectedTariff}`);
    }

    sql += template;
    setGeneratedSQL(sql);
    setCopied(false);

    if (!isExecuting) {
        setQueryResults(null);
        setExecutionMessage('');
    }
  }, [selectedProgram, configs, selectedTariff, dbConfig]);

  const copyToClipboard = () => {
    navigator.clipboard.writeText(generatedSQL);
    setCopied(true);
    setTimeout(() => setCopied(false), 2000);
  };

  if (!isConnected) {
      return (
        <div className="min-h-screen bg-slate-100 flex items-center justify-center p-4
font-sans">
            <div className="bg-white max-w-md w-full rounded-2xl shadow-xl overflow-hidden
```

```
border border-slate-200">
        <div className="bg-blue-600 p-6 text-center">
          <Database className="w-16 h-16 text-white mx-auto mb-3 opacity-90" />
          <h1 className="text-2xl font-bold text-white">SQL Exporter</h1>
          <p className="text-blue-100 text-sm mt-1">Conexión ICG / Ágora</p>
        </div>
        <form onSubmit={handleConnect} className="p-8 space-y-5">
          {connectError && (
            <div className="bg-red-50 text-red-600 p-3 rounded-lg text-sm flex
items-center gap-2 border border-red-100">
              <AlertCircle size={16} /> {connectError}
            </div>
          )}
          <div className="space-y-1">
            <label className="text-sm font-semibold text-slate-700 flex items-center
gap-2">
              <Server size={16} className="text-slate-400"/> Servidor SQL
            </label>
            <input type="text" className="w-full p-3 border border-slate-300 rounded-lg
focus:ring-2 focus:ring-blue-500 outline-none" placeholder="Ej: localhost\SQLEXPRESS"
value={dbConfig.server} onChange={(e) => setDbConfig({...dbConfig, server: e.target.value})}
/>
          </div>
          <div className="space-y-1">
            <label className="text-sm font-semibold text-slate-700 flex items-center
gap-2">
              <Database size={16} className="text-slate-400"/> Base de Datos
            </label>
            <input type="text" className="w-full p-3 border border-slate-300 rounded-lg
focus:ring-2 focus:ring-blue-500 outline-none" placeholder="Nombre BBDD"
value={dbConfig.database} onChange={(e) => setDbConfig({...dbConfig, database:
e.target.value})} />
          </div>
          <div className="grid grid-cols-2 gap-4">
            <div className="space-y-1">
              <label className="text-sm font-semibold text-slate-700 flex items-center
gap-2"><User size={16} className="text-slate-400"/> Usuario</label>
              <input type="text" className="w-full p-3 border border-slate-300
rounded-lg focus:ring-2 focus:ring-blue-500 outline-none" placeholder="sa"
value={dbConfig.user} onChange={(e) => setDbConfig({...dbConfig, user: e.target.value})} />
            </div>
            <div className="space-y-1">
              <label className="text-sm font-semibold text-slate-700 flex items-center
```

```
gap-2"><Lock size={16} className="text-slate-400"/> Password</label>
              <input type="password" className="w-full p-3 border border-slate-300
rounded-lg focus:ring-2 focus:ring-blue-500 outline-none" placeholder="••••••"
value={dbConfig.password} onChange={(e) => setDbConfig({...dbConfig, password:
e.target.value})} />
            </div>
          </div>
          <button type="submit" disabled={isConnecting} className={`w-full py-3 px-4
rounded-xl font-bold text-white shadow-lg transition-all flex justify-center items-center
gap-2 ${isConnecting ? 'bg-slate-400 cursor-wait' : 'bg-blue-600 hover:bg-blue-700'}`}>
            {isConnecting ? 'Conectando...' : <>Conectar al Motor SQL <CheckCircle
size={18}/></>}
          </button>
          <p className="text-xs text-slate-400 text-center mt-4">Versión de Escritorio -
Conexión Directa</p>
        </form>
      </div>
    </div>
   );
 }

 const CurrentIcon = configs[selectedProgram].icon;

 return (
  <div className="min-h-screen bg-slate-50 p-6 font-sans text-slate-800">
   <div className="max-w-7xl mx-auto">
    <header className="mb-8 flex justify-between items-center bg-white p-4 rounded-xl
shadow-sm border border-slate-200">
     <div className="flex items-center gap-4">
      <div className="p-2 bg-green-100 text-green-700 rounded-lg">
        <Database size={24} />
      </div>
      <div>
        <h1 className="text-xl font-bold text-slate-900">Gestor de Precios</h1>
        <div className="flex items-center gap-2 text-xs text-slate-500">
          <span className="flex items-center gap-1 text-green-600
font-semibold"><CheckCircle size={10}/> Conectado</span>
          <span>|</span>
          <span>{dbConfig.server}</span>
          <span>|</span>
          <span className="font-mono">{dbConfig.database}</span>
        </div>
      </div>
     </div>
     <div className="flex gap-3">
      <button onClick={handleDisconnect} className="flex items-center gap-2 px-4 py-2
rounded-lg bg-red-50 text-red-600 hover:bg-red-100 text-sm font-medium
transition-colors">
        <LogOut size={16} /> Desconectar
      </button>
     </div>
    </header>

    <div className="grid grid-cols-1 lg:grid-cols-12 gap-8">

     {/* COLUMNA IZQUIERDA: CONFIGURACIÓN */}
     <div className="lg:col-span-5 space-y-6">

      {/* 1. Selector Programa */}
      <div className="bg-white p-6 rounded-xl shadow-sm border border-slate-200">
       <h2 className="text-lg font-semibold mb-4 flex items-center gap-2">
        <span className="bg-blue-600 text-white w-6 h-6 rounded-full flex items-center
justify-center text-sm">1</span>
        Programa Origen
       </h2>
       <div className="grid grid-cols-1 gap-3">
        {Object.entries(configs).map(([key, conf]) => {
          const Icon = conf.icon;
          return (
            <button
              key={key}
              onClick={() => {
                setSelectedProgram(key);
                setAvailableTariffs([]); // Limpiar búsqueda al cambiar programa
                setTariffError('');
              }}
              className={`p-4 rounded-lg border-2 text-left transition-all relative
overflow-hidden flex items-center gap-4 ${
              selectedProgram === key
                ? 'border-blue-600 bg-blue-50 ring-1 ring-blue-600'
                : 'border-slate-200 hover:border-slate-300 hover:bg-slate-50'
              }`}
            >
              <div className={`p-3 rounded-full ${selectedProgram === key ?
'bg-blue-200 text-blue-700' : 'bg-slate-100 text-slate-500'}`}>
                <Icon size={24} />
              </div>
              <div>
                <div className="font-bold text-slate-800">{conf.name}</div>
                <div className="text-xs text-slate-500 mt-1">{conf.desc}</div>
              </div>
              {selectedProgram === key && (
                <div className="absolute top-0 right-0 bg-blue-600 text-white
text-[10px] px-2 py-1 rounded-bl shadow-sm">
                  ACTIVO
                </div>
              )}
            </button>
          )
        })}
       </div>
      </div>

      {/* 2. Operación / Parámetros */}
      <div className="bg-white p-6 rounded-xl shadow-sm border border-slate-200">
       <h2 className="text-lg font-semibold mb-4 flex items-center gap-2">
        <span className="bg-blue-600 text-white w-6 h-6 rounded-full flex items-center
justify-center text-sm">2</span>
        Selección de Tarifa
       </h2>

       <div className="space-y-4 animate-in fade-in">

         {/* Botón de Búsqueda */}
         <div className="flex justify-between items-center bg-slate-50 p-3 rounded-lg
border border-slate-200">
            <label className="text-sm font-semibold text-slate-700">Buscar en
BBDD:</label>
            <button
              onClick={handleSearchTariffs}
              disabled={isSearchingTariffs}
              className={`flex items-center gap-2 px-3 py-1.5 rounded-lg text-sm
font-medium transition-all ${
                isSearchingTariffs
                ? 'bg-slate-200 text-slate-500'
                : 'bg-blue-600 hover:bg-blue-700 text-white shadow-sm'
              }`}
            >
              {isSearchingTariffs ? (
                <><Loader2 size={14} className="animate-spin"/> Buscando...</>
              ) : (
                <><Search size={14}/> Buscar Tarifas</>
              )}
            </button>
         </div>

         {/* Mensaje de error de búsqueda */}
         {tariffError && (
           <div className="text-xs text-red-600 bg-red-50 p-2 rounded border border-red-100 flex items-center gap-2">
             <AlertCircle size={12}/> {tariffError}
           </div>
         )}

         {/* Lista de Tarifas o Fallback Manual */}
         <div>
           <label className="block text-sm font-bold text-slate-700 mb-2">
             {availableTariffs.length > 0 ? 'Resultados de la búsqueda:' : 'Selección Manual (ID):'}
           </label>

           {availableTariffs.length > 0 ? (
             <div className="grid grid-cols-2 gap-2 max-h-48 overflow-y-auto pr-1">
               {availableTariffs.map(tariff => (
                 <button
                  key={tariff.id}
                  onClick={() => setSelectedTariff(tariff.id)}
                  className={`p-2 rounded border text-left transition-all ${
                    selectedTariff === tariff.id
                    ? 'bg-blue-50 border-blue-500 ring-1 ring-blue-500'
                    : 'bg-white border-slate-200 hover:border-slate-300'
                  }`}
                 >
                   <div className="font-bold text-slate-800 text-xs truncate">{tariff.nombre}</div>
                   <div className="text-[10px] text-slate-500">ID: {tariff.id}</div>
                 </button>
               ))}
             </div>
           ) : (
             <div className="flex gap-3">
               {[1, 2, 3, 4].map(num => (
                 <button key={num} onClick={() => setSelectedTariff(num)}
 className={`flex-1 py-3 rounded-lg font-bold border-2 transition-all ${selectedTariff ===
 num ? 'border-blue-600 bg-blue-600 text-white' : 'border-slate-200 bg-white
 text-slate-400 hover:border-blue-400'}`}>
                   {num}
                 </button>
               ))}
             </div>
           )}
         </div>

         {/* Confirmación visual */}
         <div className="text-xs text-center text-slate-400 border-t border-slate-100
 pt-2">
           Tarifa ID seleccionada: <strong>{selectedTariff}</strong>
         </div>
       </div>
     </div>

     {/* BOTÓN DE EJECUCIÓN */}
     <button
       onClick={handleExecuteSQL}
       disabled={isExecuting}
       className={`w-full py-4 rounded-xl shadow-lg font-bold text-lg flex items-center
 justify-center gap-3 transition-all transform active:scale-95 ${
         isExecuting
         ? 'bg-slate-300 text-slate-500 cursor-not-allowed'
         : 'bg-gradient-to-r from-emerald-500 to-emerald-600 text-white
 hover:from-emerald-600 hover:to-emerald-700'
       }`}
     >
       {isExecuting ? (
         <>
            <Loader2 className="animate-spin" size={20} />
            Cargando Datos...
         </>
       ) : (
         <>
            <Play fill="currentColor" size={20} />
            Ejecutar SQL
         </>
       )}
     </button>
```

```
     </div>

     {/* COLUMNA DERECHA: SQL Y RESULTADOS */}
     <div className="lg:col-span-7 flex flex-col gap-6">

      {/* PANEL SQL */}
      <div className="bg-slate-900 text-slate-300 rounded-xl shadow-lg flex flex-col
min-h-[300px]">
       <div className="bg-slate-800 p-3 border-b border-slate-700 flex justify-between
items-center">
        <span className="font-mono text-sm font-semibold text-white flex gap-2"><Code
size={18}/> Consulta SQL Generada</span>
        <button onClick={copyToClipboard} className="text-xs bg-blue-600
hover:bg-blue-500 text-white px-3 py-1 rounded flex gap-1 transition-colors">
          {copied ? <CheckCircle size={14}/> : <Copy size={14}/>} {copied ? 'Copiado' :
'Copiar'}
        </button>
       </div>
       <div className="p-4 flex-grow font-mono text-xs overflow-auto max-h-[400px]">
        <pre className="whitespace-pre-wrap break-all text-emerald-400
leading-relaxed">
          {generatedSQL}
        </pre>
       </div>
      </div>

      {/* PANEL DE RESULTADOS / MENSAJES */}
      {executionMessage && (
        <div className="p-4 rounded-xl border flex items-start gap-3 animate-in
slide-in-from-bottom-2 bg-blue-50 border-blue-200 text-blue-800">
          <CheckCircle className="mt-0.5 flex-shrink-0" size={20}/>
          <div className="flex-grow">
            <p className="font-medium">{executionMessage}</p>
          </div>
        </div>
      )}

      {/* TABLA DE DATOS */}
      {queryResults && queryResults.length > 0 && (
        <div className="bg-white rounded-xl shadow-md border border-slate-200
overflow-hidden animate-in fade-in">
          <div className="bg-slate-50 p-4 border-b border-slate-200 flex justify-between
items-center">
```

```
            <div>
              <h3 className="font-bold text-slate-700 flex items-center gap-2">
                <TableIcon size={18} className="text-slate-500"/>
                Vista Previa {selectedProgram === 'agora' ? '(Hostelería)' : '(Retail)'}
              </h3>
              <p className="text-xs text-slate-400 mt-1">Mostrando los primeros 100 de
{queryResults.length} registros.</p>
            </div>
            <button
              onClick={handleExportExcel}
              className="bg-green-600 hover:bg-green-700 text-white px-4 py-2
rounded-lg text-sm font-medium flex items-center gap-2 shadow-sm transition-colors"
            >
              <FileSpreadsheet size={18} />
              Exportar Todo (.xlsx)
            </button>
          </div>
          <div className="overflow-x-auto max-h-[400px]">
            <table className="w-full text-sm text-left">
              <thead className="bg-slate-100 text-slate-600 font-semibold uppercase
text-xs sticky top-0 shadow-sm">
                <tr>
                  {Object.keys(queryResults[0]).map((header) => (
                    <th key={header} className="px-4 py-3 border-b border-slate-200
whitespace-nowrap">
                      {header}
                    </th>
                  ))}
                </tr>
              </thead>
              <tbody className="divide-y divide-slate-100">
                {queryResults.slice(0, 100).map((row, idx) => (
                  <tr key={idx} className="hover:bg-slate-50 transition-colors">
                    {Object.values(row).map((val, cellIdx) => (
                      <td key={cellIdx} className="px-4 py-3 whitespace-nowrap
text-slate-700">
                        {val}
                      </td>
                    ))}
                  </tr>
                ))}
              </tbody>
            </table>
          </div>
          <div className="bg-slate-50 p-2 text-xs text-center text-slate-400 border-t border-slate-200">
            {queryResults.length > 100 ? (
              <span className="font-semibold text-amber-600">Vista previa limitada. Exporta a Excel para ver el total de {queryResults.length} filas.</span>
            ) : (
              <span>Mostrando {queryResults.length} registros.</span>
            )}
          </div>
        </div>
      )}

     </div>

    </div>
   </div>
  </div>
 );
};

export default App;
```