

Name:	FR #1 Acceder al programa
Summary:	El programa permitirá almacenar, leer y agregar información suministrada
Inputs:	-Valor numérico en que efectuara dicha acción: 1. Insertar comando 2. Importar datos desde archivo .SQL 2. Salir -
Outputs:	Dato guardado, leído o eliminado exitosamente.

Name:	FR #2 Agregar País
Summary:	Se creará una base de datos para los países almacenados.
Inputs:	-Valor numérico para la ejecución de esta acción (1), posteriormente colocar el comando INSERT INTO.  -Strings o valores numéricos para crear un dato y almacenarlo: <b>Id.</b> De tipo String. Debe ser generado mediante UUID <b>Name.</b> De tipo String. Es el parámetro con el nombre de la ciudad <b>Population.</b> De tipo Double. Almacena la población actual de la ciudad en millones de personas <b>CountryCode.</b> De tipo String. Representa el indicativo del país
Outputs:	<ul style="list-style-type: none"> <li>• Correcto: País guardado exitosamente.</li> <li>• ID incorrecto: ID incorrecto (Excepción).</li> <li>• Formato incorrecto: Si el dato es un String agregar comilla simple, si es numérico agregarlo sin comillas.</li> </ul>

Name	FR #3 Agregar Ciudad
Summary:	Este va asociado al País una vez agregado
Inputs:	-Valor numérico para la ejecución de esta acción (1), una vez creado se debe colocar el comando INSERT INTO para añadir una ciudad.  - Strings o valores numéricos para crear un dato y almacenarlo: <b>Id.</b> De tipo String. Debe ser generado mediante UUID <b>Name.</b> De tipo String. Es el parámetro con el nombre de la ciudad <b>CountryID.</b> De tipo String. Contendrá el identificador del país <b>Population.</b> De tipo Integer. Almacena la población actual de la ciudad
Outputs:	<ul style="list-style-type: none"> <li>• Correcto: Ciudad guardado exitosamente.</li> <li>• ID incorrecto: ID incorrecto (Excepción).</li> </ul>

	<ul style="list-style-type: none"> <li>• Formato incorrecto: Si el dato es un String agregar comilla simple, si es numérico agregarlo sin comillas.</li> </ul>
--	--

Name	FR #4 Búsqueda en base de Datos
Summary:	El usuario podrá acceder a la base de datos mediante unos comandos previamente escritos para filtrar la información.
Inputs:	<p>-Valor numérico para la ejecución de esta acción (1), posteriormente debe filtrar los datos con el comando SELECT + FORM + (dato a averiguar dado el parametro).</p> <p>-Comando para filtraciones:</p> <ul style="list-style-type: none"> <li>• Example: <code>SELECT * FROM countries WHERE population &gt; 100</code></li> <li>• Example: <code>SELECT * FROM countries WHERE population &lt; 30</code></li> <li>• Example: <code>SELECT * FROM countries WHERE population = 60</code></li> </ul> <p>Nota:</p> <ul style="list-style-type: none"> <li>• <b>Azul</b>: Parámetro para la filtración de datos. (puede variar según lo que se desee averiguar.</li> <li>• <b>Verde</b>: Signo de comparación para la filtración de datos (=, &lt;, &gt;).</li> </ul>
Outputs:	<ul style="list-style-type: none"> <li>• Si hay datos: Filtración exitosa.</li> <li>• Si no hay datos: No hay datos que mostrar.</li> </ul>

Name	FR #5 Ordenamiento de datos
Summary:	El programa filtrara, ordenara y mostrara los datos que el usuario desee visualizar.
Inputs:	<p>- Valor numérico para la ejecución de esta acción (1), posteriormente ordenara los datos con el comando SELECT + FROM + parámetro a ordenar + WHERE + parámetro a usar + ORDER BY+ parámetro a hallar/ordenar.</p> <p>-Comando para ordenamiento:</p> <ul style="list-style-type: none"> <li>• Example: <code>SELECT * FROM countries WHERE population &gt; 100 ORDER BY name</code></li> <li>• Example: <code>SELECT * FROM cities WHERE name = 'Guadalajara' ORDER BY population</code></li> </ul> <p>NOTA:</p> <p><b>LOS PARAMETROS INIICALES PUEDEN SER:</b> countries, cities.</p>

	<p>LOS PARAMETOS SECUNDARIOS PUEDEN SER: id, name, population, countryCode, countryID.</p> <p>LOS PARAMETROS DE ESTE TIPO SON LIBRE TENIENDO EN CUENTA QUE DEBEN ESTAR DENTRO DE LA BASE DE DATOS.</p>
Outputs:	<ul style="list-style-type: none"> <li>Mostrara los datos ordenados por el parámetro a seguido del comando ORDER BY previamente con el comando SELECT.</li> </ul>

Name	FR #6 Eliminación de datos
Summary:	Se permitirá eliminar un registro
Inputs:	<p>- Valor numérico para la ejecución de esta acción (1), posteriormente debe hacerse uso del comando DELETE.</p> <ul style="list-style-type: none"> <li>Example: DELETE FROM cities WHERE country = 'Colombia'</li> <li>Example: DELETE FROM cities WHERE population &gt; 50</li> </ul>
Outputs:	

Name	FR #7 Importar archivo .SQL
Summary:	El programa permitirá almacenar datos externos para su uso
Inputs:	- Valor numérico para la ejecución de esta acción (2)
Outputs:	