

ALFACONVERTER

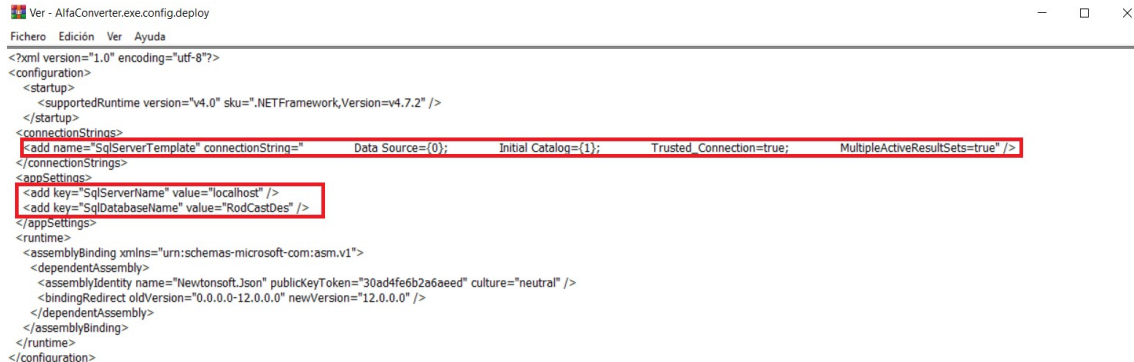
Manual de usuario



KYBELE

Configuración inicial

En el directorio de la aplicación existe un fichero llamado “**AlfaConverter.exe.config**”, cuyo contenido permite configurar algunos parámetros básicos. Los más importantes son:



```
Ver - AlfaConverter.exe.config.deploy
Fichero Edición Ver Ayuda
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.7.2" />
  </startup>
  <connectionStrings>
    <add name="SqlServerTemplate" connectionString="Data Source={0}; Initial Catalog={1}; Trusted_Connection=true; MultipleActiveResultSets=true" />
  </connectionStrings>
  <appSettings>
    <add key="SqlServerName" value="localhost" />
    <add key="SqlDatabaseName" value="RodCastDes" />
  </appSettings>
  <runtime>
    <assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">
      <dependentAssembly>
        <assemblyIdentity name="Newtonsoft.Json" publicKeyToken="30ad4fe6b2a6aeed" culture="neutral" />
        <bindingRedirect oldVersion="0.0.0.0-12.0.0.0" newVersion="12.0.0.0" />
      </dependentAssembly>
    </assemblyBinding>
  </runtime>
</configuration>
```

Ilustración 1: Campos importantes para la configuración de AlfaConverter

configuration/connectionStrings

Contiene una cadena de conexión de base de datos con nombre “**SqlServerTemplate**”, que es utilizado por la aplicación para conectar a un servidor SQL Server (parámetro {0}) y a una base de datos (parámetro {1}).

configuration/appSettings

Contienen configuraciones de tipo clave-valor. Hay dos:

- **SqlServerName:** Nombre del servidor al que se conecta por defecto al iniciar.
- **SqlDatabaseName:** Nombre de la base de datos a la que se conecta por defecto al iniciar

Uso de la aplicación

Al abrir la aplicación, inicialmente se conecta a la base de datos del servidor configurado en el fichero App.config. Por ejemplo, ver la siguiente figura:

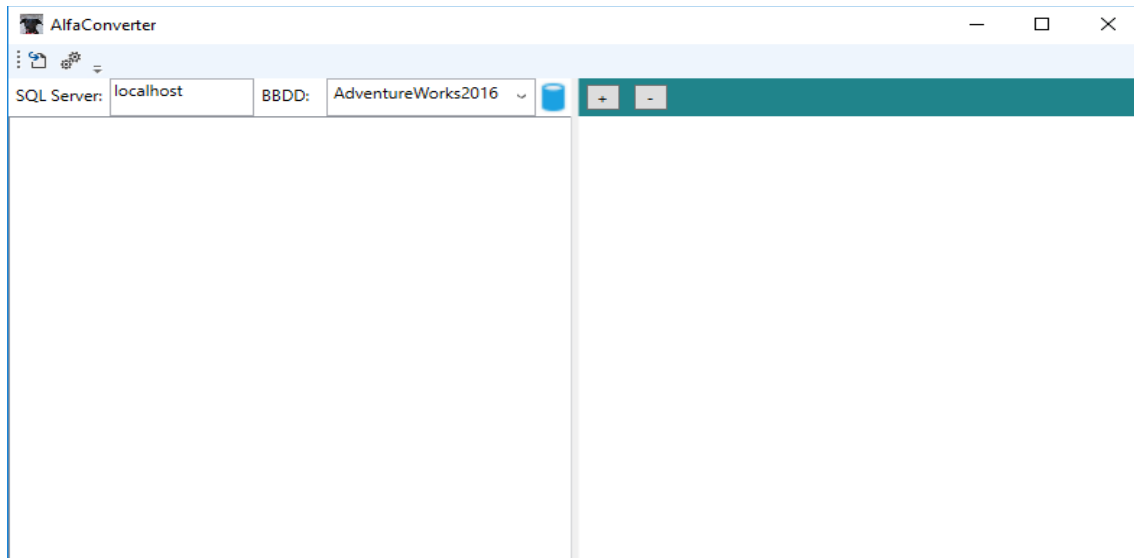


Ilustración 2: Apertura inicial de la aplicación

Dado que en el proceso de conexión realiza un análisis de la base de datos en busca de las tablas y sus columnas, puede tomar un tiempo en que acabe por mostrar en el panel izquierdo el resultado. Dicho resultado se puede apreciar a continuación:

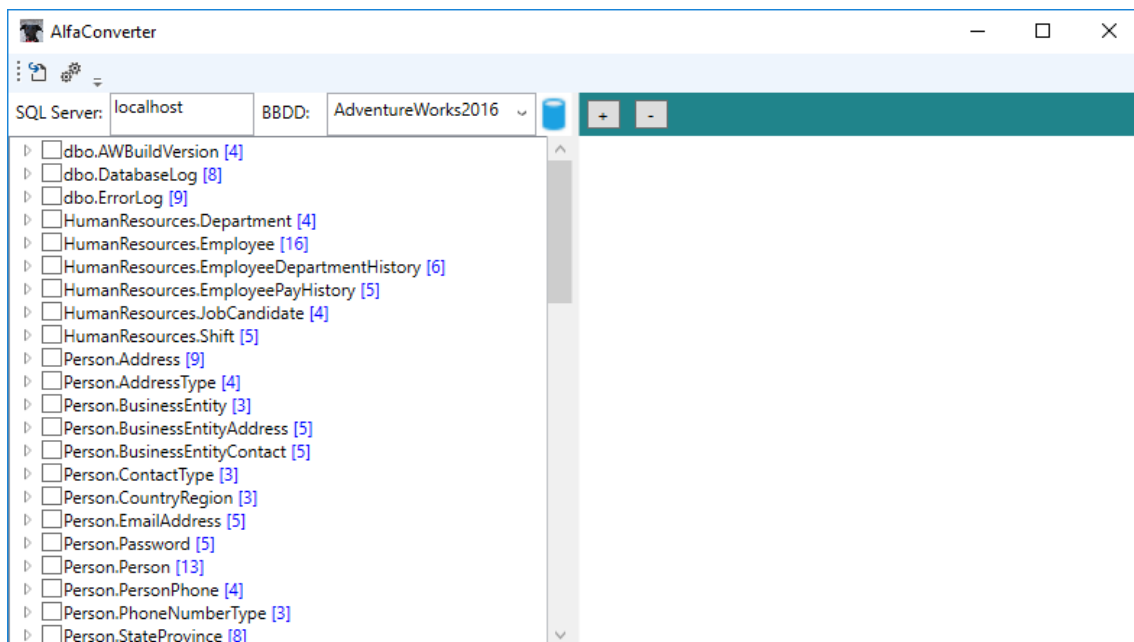


Ilustración 3: Carga inicial de las tablas de la base de datos configurada

En este estado, el usuario puede modificar el nombre del servidor, en cuyo caso la lista desplegable de bases de datos se actualiza con el nuevo listado. El usuario también puede seleccionar otra base de datos si así lo desea:

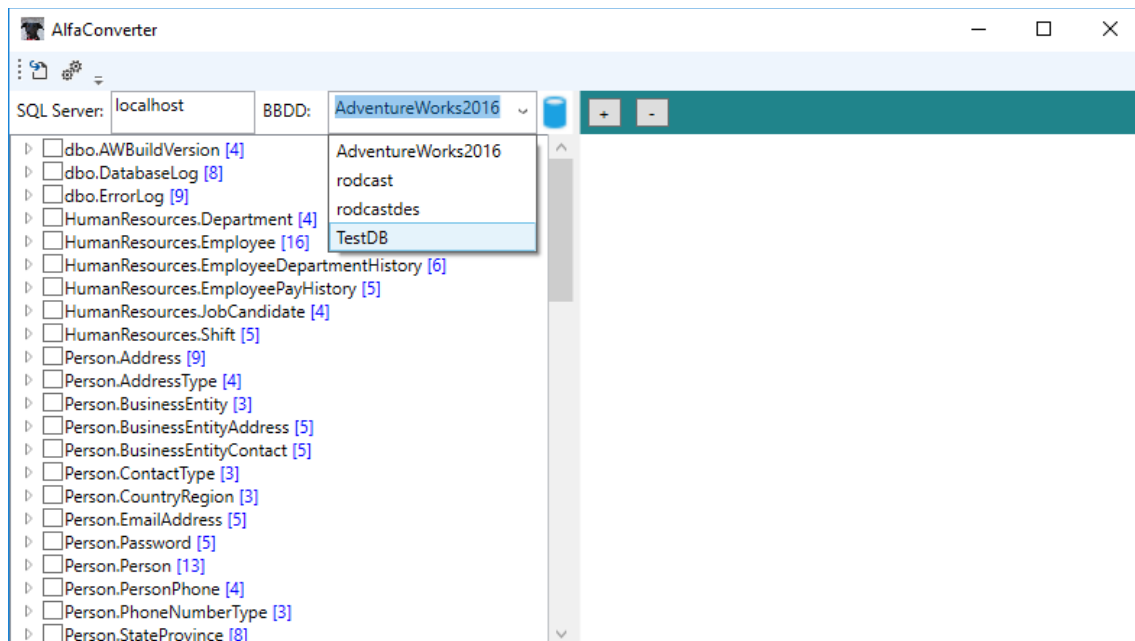


Ilustración 4: Desplegando el listado de BD

Al seleccionar una base de datos diferente, el listado de tablas del panel izquierdo se actualiza como se puede apreciar a continuación:

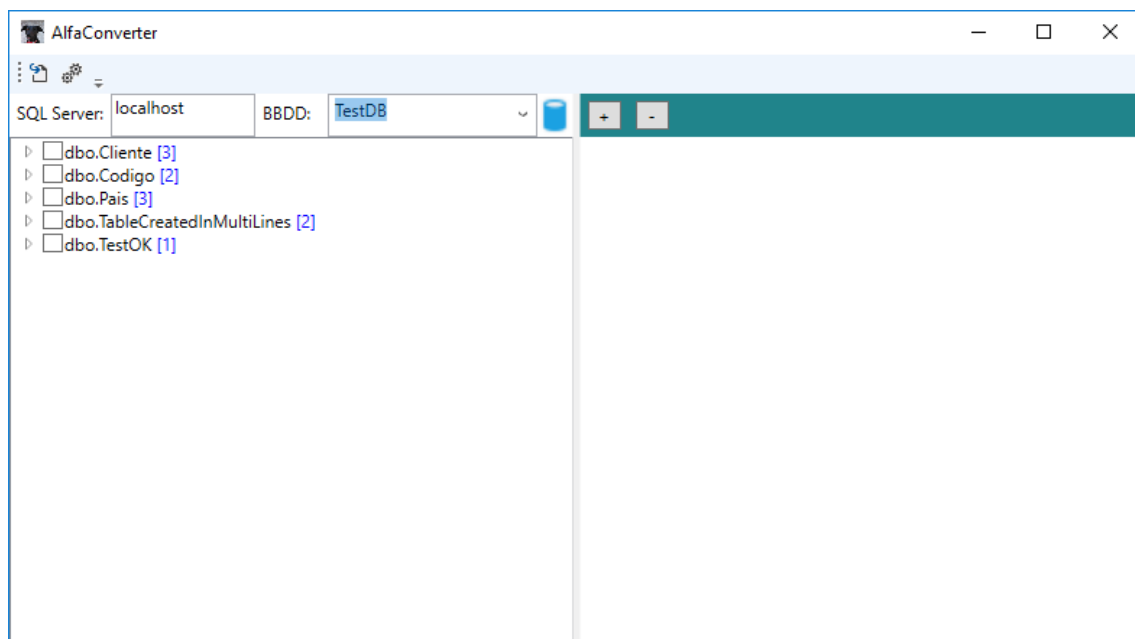


Ilustración 5: Selección de otra base de datos

El panel izquierdo permite realizar una exploración ágil de la base de datos a través de sus tablas. Desplegar una tabla muestra su listado de columnas:

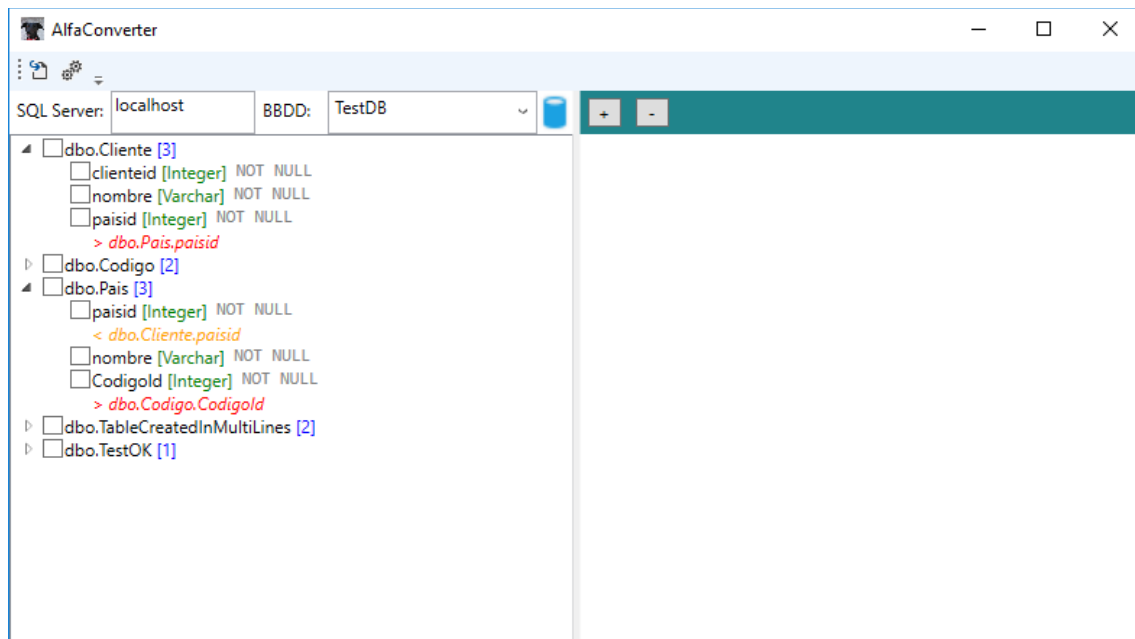


Ilustración 6: Mostrando listado de columnas en tablas

En la anterior figura se puede observar que además del listado de columnas, también se muestra información adicional diferenciada por un código de colores:

- Verde: Tipo de la columna.
- Gris: Indica si el campo puede ser NULL.
- Rojo: Indica hacia qué tabla apunta una clave foránea.
- Naranja: Indica el listado de tablas que apuntan al campo de la tabla actual.

Conversión de SQL a JSON

Siendo fácil obtener información importante de la tabla Pais, se va a proceder a procesar dicha tabla a JSON detallando los campos que posee:

- Posee tres campos: paisid, nombre y Codigold. También se puede ver rápidamente sus tipos y si pueden ser NULL.
- El campo Pais.Codigold apunta a la columna Codigo.Codigold.
- El campo Pais.paisid es referido por la columna Cliente.paisid.

A continuación, el usuario puede seleccionar una o más tablas:

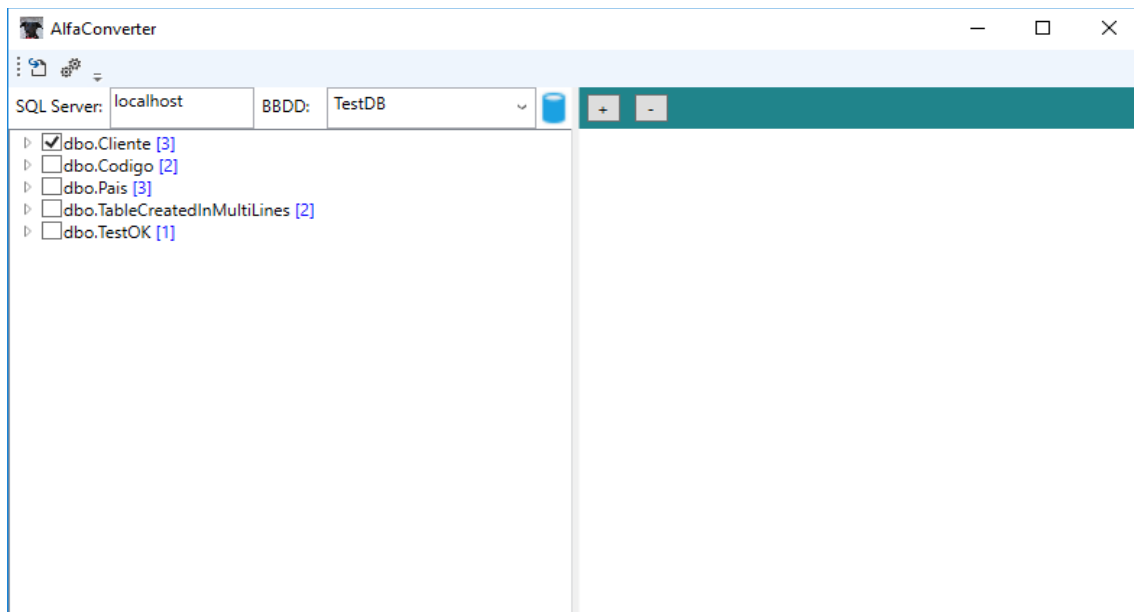


Ilustración 7: Selección de una tabla SQL

Al hacer click en el icono de procesar (Engranaje):

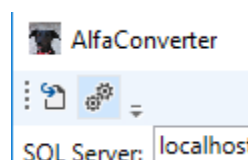


Ilustración 8: Botón de "Procesar" en la herramienta AlfaConverter

La aplicación realiza un análisis de la tabla seleccionada y genera un JSON equivalente en el panel derecho:

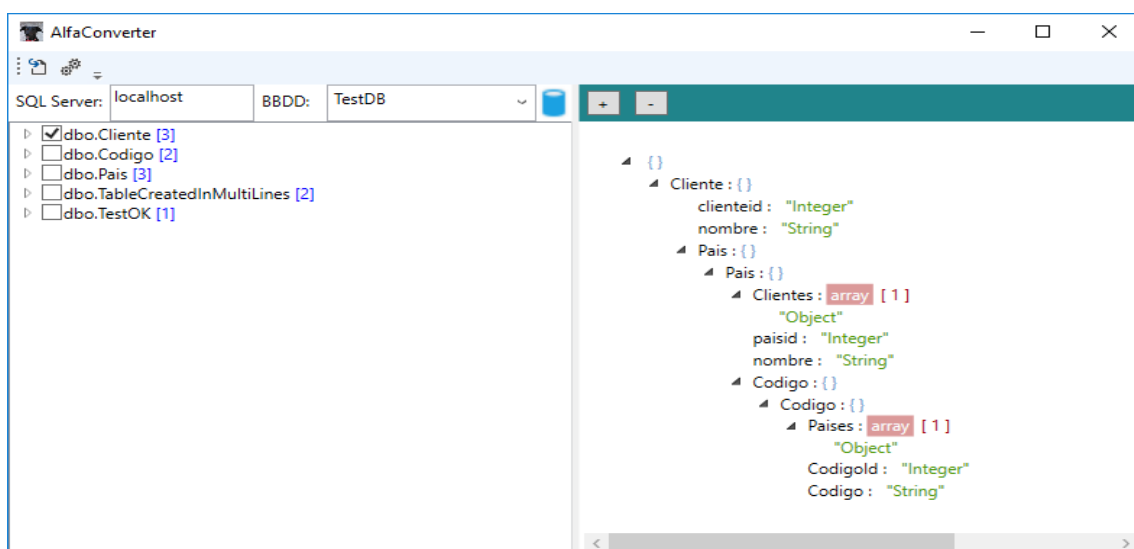


Ilustración 9: Conversión de una tabla SQL a JSON

Como se puede observar en la ilustración 8, el JSON que se ha generado explica las relaciones SQL con origen en la tabla Cliente:

- Como Cliente.paisid es una clave foránea hacia la tabla Pais, en la estructura JSON se traduce en un campo de nombre “Pais” que a su vez contiene campos según su tabla.
- Como Pais.Codigold es una clave foránea hacia la tabla Codigo, en la estructura JSON se traduce en un campo de nombre “Codigo” dentro del campo “Pais” que a su vez contiene campos según su tabla.
- Como Pais.paisid es referido por la tabla Cliente, se crea un campo “Clientes” (nótese la automatización del plural), que es de tipo array. Como el objeto “Cliente” ya fue definido en un nivel superior, para evitar bucles infinitos, se frena aquí esta rama de la estructura JSON¹⁰.
- Ocurre lo mismo con el campo “Paises” autogenerado en plural dentro del campo “Codigo”. Como la estructura “Pais” ha sido generada en niveles superiores, también se frena aquí la definición de la rama para evitar el bucle infinito.
- Los tipos SQL se convierten a tipos NoSQL:
 - Integer se traduce a Integer.
 - Varchar se traduce a String.

Conversión de varias tablas SQL a JSON

También es posible generar un JSON de múltiples tablas como se aprecia en la siguiente imagen:

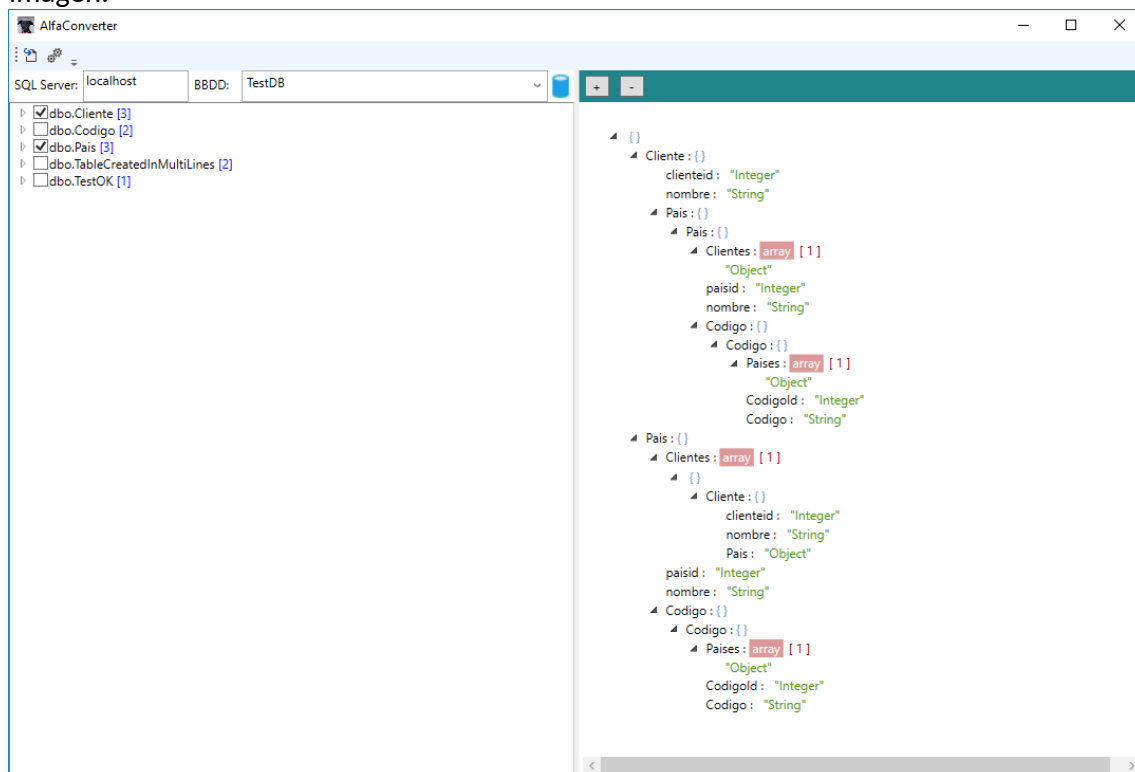


Ilustración 10: Conversión de dos tablas SQL a estructura JSON

En este caso, la estructura para la tabla Pais se genera siguiendo las relaciones tomando como punto de partida dicha tabla. Al generar conjuntamente Cliente y Pais, es fácil darse cuenta de la acción de la protección de bucles infinitos:

- El campo “Pais” frena su definición en el campo “Clientes” porque ya se definió en un nivel superior.

- Sin embargo, cuando se define la estructura “Pais” de forma paralela, la estructura “Clientes” sí es definida.

Importación de un script SQL

Además de poder acceder a una base de datos ya existente, la aplicación dispone de la posibilidad de importar un script SQL con sentencias DDL mediante el botón de importar como se aprecia en la siguiente imagen:

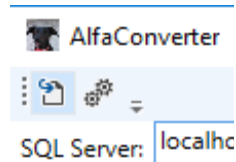


Ilustración 11: Botón "Importar" de la herramienta AlfaConverter

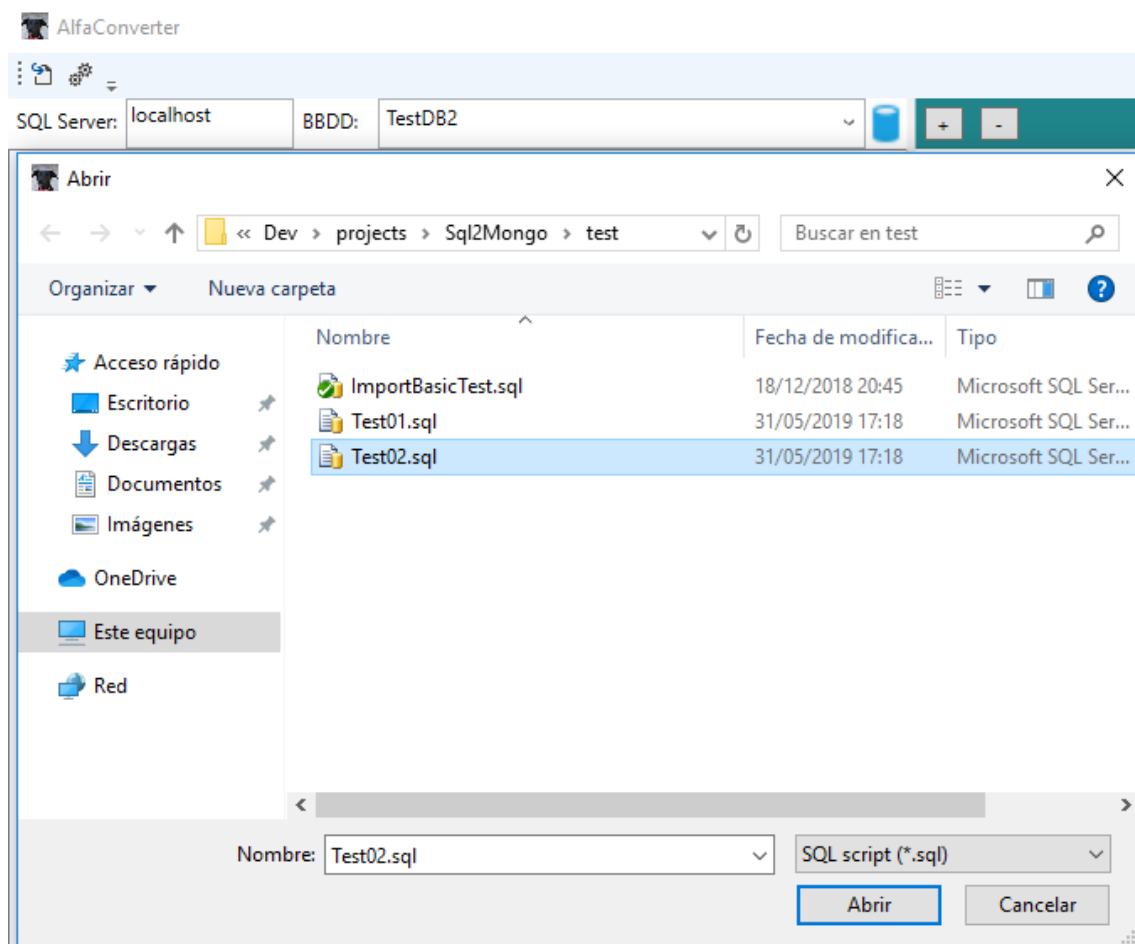


Ilustración 12: Importación de un script SQL en AlfaConverter

Si la importación ha sido realizada correctamente, la aplicación mostrará un mensaje similar al que aparece a continuación como notificación para el usuario:

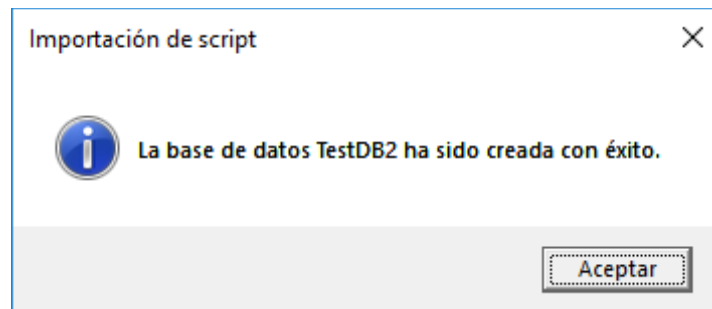


Ilustración 13: Mensaje de notificación por parte de la herramienta AlfaConverter

Tras lo cual, es posible seleccionar la nueva base de datos como una más:

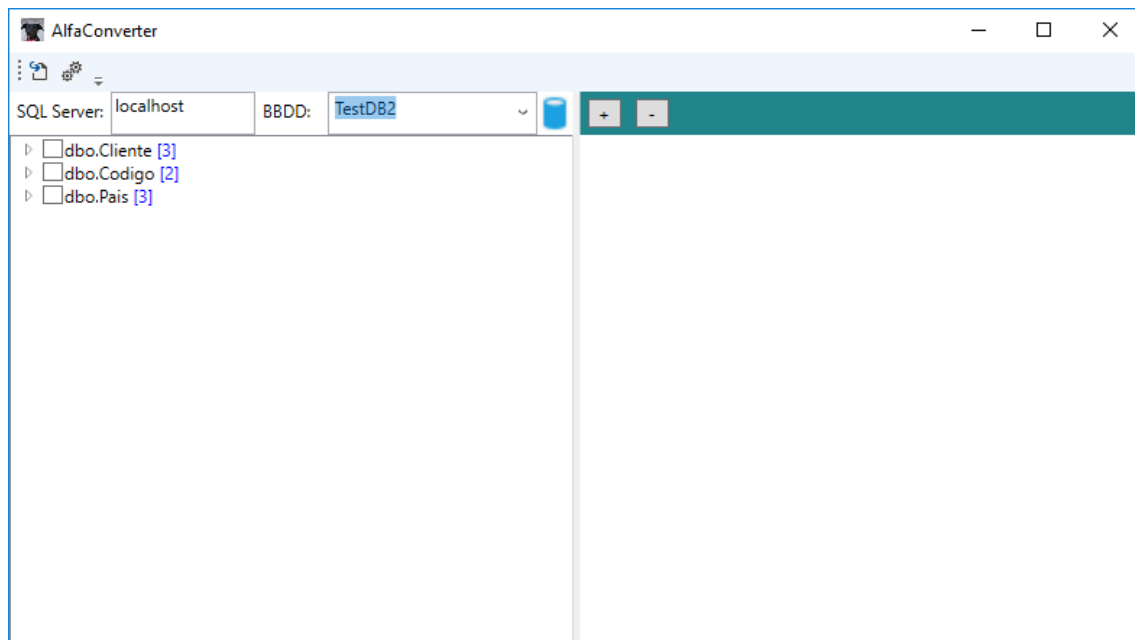


Ilustración 14: Visualización de una nueva BD de datos importada