En vez de tomar de un HTML toma de un XML

Lo que está dentro de <channel> son *items*

[www.w3schools.com/xml/xml\_rss.asp](http://www.w3schools.com/xml/xml_rss.asp)

* En el Ejemplo del Moodle, hizo la PublishingDate nullable (xq los objetos DateTIme no pueden ser nulables en gral) 🡪 algo asi
* El logger me permite ir dejando pistas durante la aplicación🡪 para registrar todos los errores que van sucediendo, o datos importates🡪 ya que el usuario, cuando le vas a preguntar qué le tiró el error, en gral no te sabe ni decir. Xej, guardo el tiempo qe tarda en acceder al archivo rss.
* Le pongo los loggers donde creo que puede fallar
* Lo escribe en un archivo, en una base de datos, donde sea que le hayamos indicado
* La clase XMLTextReader es para leer XMLs
* cLogger.DebugFormat🡪 lo resuelve haya o no haya debug.
* cLogger.Info🡪 es informativo
* GetXmlNodeValue<TResult> () 🡪 le puso TResult genérico para que sirva para todos.
* La clase SyndicationFeedRssReader.cs es otra variabte a la RawXmlRssReader.cs. Se apoya en la librería System.ServiedModel. Syndication🡪 es más simple xq el Syndication la abstrae y simplifica
* En la interfaz (FrmMain.cs), hay un contenedor de IOC (bwRssReader\_DoWork), que me resuelve la abstracción (no entendí bien)
  + 🡪 es el trabajo que tiene que realizar
  + 🡪 el RunWorkerCompleted es para cuando terminó el trabajo de DoWork.
* El RunWorkerAsync es para cuando la interfaz es muy pesada, que no se congele la pantalla cargando los datos.
* Fijarse en el App.config!! Lo modificó🡪 agregó dos secciones : la de “unity” y la de “logging”