De la clase JobLoader.

* El método LoadMessage tiene un nombre de parámetro repetido (message) hay que renombrarlo.
* Las variables “l” y “t” no están inicializadas.
* La conexión a la base de datos está siendo manejado en el misma clase. Se sugiere usar la arquitectura de 3 niveles( presentación, negocio y acceso a datos).
* En la capa de datos el transact debería estar en un store procedure para aprovechar su plan de ejecución.
* En la capa de datos la conexión debería estar dentro de la instrucción “using” para así al salir de la instrucción se cierre la conexión a base de datos. Actualmente no se está cerrando la conexión.
* El método de acceso a datos debe mandar el tipo y tamaño correcto por cada parámetro.
* En la capa de negocio se tendría toda la lógica menos el llamado a la base de datos.
* Se necesita una de entidades para comunicar los niveles mediante objetos relevantes, por ejemplo a capa de datos para registrar el error se puede enviar una entidad Error.
* En la capa de negocio no se está programado el flujo de registrar en base de datos o en archivo o en consola, sino que se ejecutan los tres a la vez. Se sugiere agregar una instrucción “if” para cada caso de tipo de log.
* En la capa de negocio en el log por archivo, no hay forma de markar el error como error, mensaje o warning. Se sugiere incrementar el mensaje de error con una trama que indique el tipo de error.
* En la capa de negocio, los mensajes debe estar en un archivo de recursos no en duro.
* En la capa de negocio los valores de “t” debe figurar en constantes o un enumerador en la capa de entidades.
* En la capa de negocio, en la sección de registro de archivos, se debe validar si el archivo existe.
* El código tiene sección de validación, proceso y resultado, se sugiere hacer pruebas unitarias con Nunit para validar si se pueden hacer los tres tipos de registros, si se están haciendo las validaciones de la sección validaciones.