# **Arquitectura de la solución**

Para la arquitectura de la solución, se tuvo en cuenta un sistema distribuido, el cual presenta la siguiente consiguración:

Dos Workers en background, los cuales tienen las siguientes responsabilidades.

* Worker EnviadorPost. Encargado de leer los post generados de forma aleatoria y transmitirlos al servicio de procesamiento de los post basado en WebApi.
* Worker GeneradorPost. Encargado de la generación masiva de los datos de los post, este persiste en base de datos (Relacional) la información de los post generados.

Un servicio de procesamiento de post basado en WebApi:

* Denominado Post.Servicios.Web, el servicio web basado en WebApi, es capaz de recibir los post y persistirlos en una base de datos que permite el análisis de los datos que llegan.

Un Worker analizador de los post.

* Worker AnalizadorPosts. Encargado del análisis de los post que son persistidos.

Los Worker se definen con el fin de que se pueda tener procesamiento desatendido y que no se afecten algunas operaciones de procesamiento de los post. El Worker de envío masivo, recupera de a 1000 solicitudes o post generados y persistidos en una sola ejecución, y este es capaz de transmitirlos al servicio basado en WebApi.

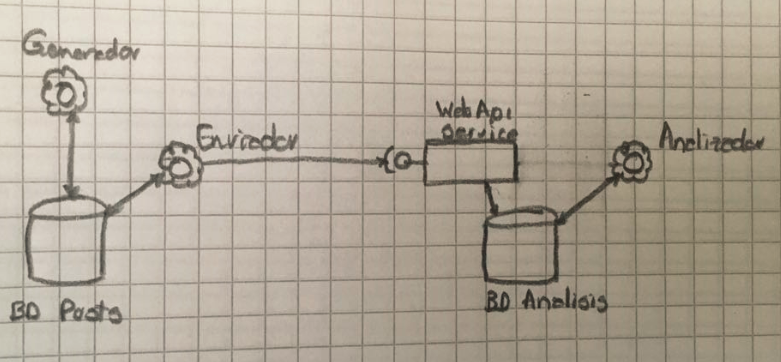


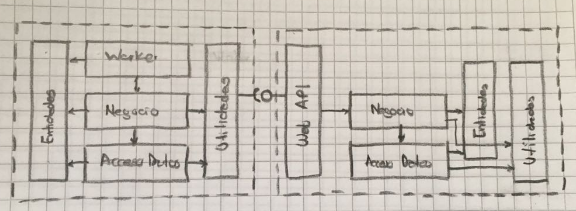
Ilustración 1. Arquitectura general

## **Persistencia**

La persistencia de a solución, consiste en una base de datos SQL Server, con una estructura de datos común de tablas. Los datos son persistidos en forma de columnas, y una serie de procedimientos almacenados se encargan de la administración de la información de los posts.

## **Arquitectura de aplicación**

De forma general, la arquitectura de las aplicaciones emplea un esquema multicapa sencillo y acceso a Datos basado en tecnología ADO.NET, debido a que se requiere mantener control sobre la liberación de recursos, en el consumo del pool de conexiones al motor de Base de datos.

Ilustración 2. Arquitectura general de la aplicación.