**Funciones esenciales de Gulp**



Las siguientes son funciones que trae la librería Gulp y que nos facilitan la programación:

**src(‘url-file’ o [‘url-file’,’url-file’]):**

Esta función devuelve un archivo o una lista de archivos en formato blob (un formato manejado por Javascript). Esta función recibe como parámetros un listado o array de string, que vienen siendo direcciones de archivos o patrones para identificar archivos. Además, nos ayuda a obtener los archivos que vayamos a procesar, sea el caso de procesar hojas de estilos o templates como jade, entre otros ejemplos.

**dest(‘’):**

Esta función no facilita el traslado de los archivos procesados por gulp, realmente aplica como un especie de return o respuesta final de cada tarea, aunque realmente no siempre devolvamos archivos físicos procesados. Nos permite dar un lugar en específico a los archivos que acaban de ser tratados, según las indicaciones que hayamos dado a nuestra tarea.   
Un ejemplo muy claro del uso de esta función es cuando procesamos un archivo de algún preprocesador de estilos ya sea Stylus, Sass, Less y queremos darle destino a nuestro archivo resultado, dentro de la función **.pipe(gulp.dest(‘./build/css’)).** Este sería un ejemplo claro de la forma en que utilizamos esta función.

**task(‘’, [‘’], fn):**

Esta función nos permite definir nuestras tareas o una operación a automatizar. El primer parámetro que recibe esta función es el nombre que le vamos a asignar a esta tarea. Este nombre debe ir entre comillas ‘nombre fn’, seguido de este parámetro que es opcional, es un listado o un array de tareas ya definidas, que se quiere que se realicen antes de ejecutar nuestra tarea que hemos definido, ejemplo, [‘compitarjs’,’compilarcss’] entre otros. Y en nuestro último parámetro que lo he definido como **fn o función,** asignamos una función anónima, que la definimos normalmente de la siguiente manera, function(){}, de esta forma podemos definir nuestras tareas.

**watch(‘url-files or url-file’, opts, [task]/fn):**

Esta función nos permite disparar una función, tan pronto el blob que le hemos pasado en el primer parámetro haya sido editado, eliminado, renombrado o agregado. Esta función es muy útil ya que nos ayuda a automatizar, ya sea una tarea o un conjunto de tareas.  
  
Ya que hemos conocido las funciones básicas de gulp, podemos aplicar estas funciones en tareas más comunes en el desarrollo de aplicaciones que realizamos diariamente, tales como concatenar archivos javascript o poder realizar un transpiler de archivos escritos en Ecma6, minificar archivos js o css, procesar hojas de estilos escritas ya sea en Stylus, Less o Sass, entre muchas otras tareas que encontramos en el diario vivir de un desarrollador o programador front-end.