Package System

Al iniciar REPL podemos cambiar de modo para acceder al administrador de paquetes con `]`. El modo Pkg nos indica el *environment* en que nos encontramos. El ambiente por default viene con la instalación de Julia.

Cambiar de environment

Para cambiar de ambiente utlizamos los siguientes comandos. Para activar el directorio actual (donde se inició el REPl) utlizamos:

```
activate.
```

Para activar una ruta específica utilizamos:

```
activate path
```

Para activar el ambiente por default:

```
activate
```

Comandos básicos

add

Para añadir un package se utiliza

```
add PACKAGE
```

Se añade un paquete en la sesión.

> 💡Cuando diferentes proyectos utilizan la misma versión de un paquete, tal paquete solo se almacena una vez en el disco. El comando `add` **descarga y precompila el paquete, si no**

existía previamente, y registra el paquete como una dependencia del ambiente actual.

`st`

El comando `st` (status) muestra los paquetes instalados.

```
st
```

`up`

Para actualizar los paquetes instalados se utiliza `up`.

```
up
```

`rm`

El comando `rm` se utiliza para remover un paquete instalado.

```
rm PACK1 ...
```

> 💡 El comando elimina el paquete de la lista de dependencias del ambiente, pero no elimina los archivos. El *garbage collector* automático de Julia se encarga de eliminar archivos que no sean dependencias de otro proyecto y no se hayan utilizado en alrededor de 30 días.

Naturaleza de un paquete

Un paquete es en realidad un módulo de Julia asociado con un *Project.toml* file. El archivo que los contiene se despliega como sigue:

```
Project.toml

src > SomePackage.jl
```

Junto al archivo `.toml` se encuentra un directorio `src` con un archivo `.jl` nombrado como el paquete. En el archivo se define un módulo con el nombre del paquete:

```
module SomePackage
...
end
```

Asimismo, tal estructura se suele almacenar en un directorio con el nombre del paquete, pero no es estrictamente necesario.

Se puede pensar en un paquete como un ambiente con un módulo dentro.

Environment

Un environment es en realidad un lugar que almacena un archivo `Project.toml` y `Manifest.toml`. El `Project.toml` debe contener los siguientes campos:

```
name = ...
uuid = ... # Identificador
authors = ...
version = ...
```

Conforme se añadan dependencias al ambiente con el comando `add` se crea `Manifest.toml` que contiene el grafo con todas las dependencias.

Load Path

Al usar `import` o `using` Julia busca los paquetes en ciertas direcciones definidas por el vector:

```
LOAD_PATH
```

La traducción para estas notaciones se obtiene con la función del módulo `Base`:

```
Base.load_path()
```