Librerías, Framework y Paquetes Nuget

.

## 1.- librerías.

En el universo de programación existe algo que se llama “*librerías*”.

***¿Que son o que hacen las librerías?***

* Las “*librerías*” almacenan funciones y clases predefinidas a las cuales nosotros podemos invocar después de la instalación de la librería.
* Las “*librerías*” fueron construidas para facilitar el trabajo de los desarrolladores e impulsar el proceso de desarrollo.

Las librerías pueden ser “*OPEN SOURCE*” o “*CLOSED SOURCE*”

***¿Que son librerías “OPEN SOURCE”?***

“OPEN SOURCE” se significa que el código fuente está disponible públicamente para cualquiera que lo desee, y los programadores pueden leer o cambiar ese código si lo desean.

Ventajas:

* Por lo general es gratuito.
* Colaboración Publica (El público puede corregir errores, agregar funciones y mejorar el rendimiento en un período de tiempo relativamente corto)

Desventajas:

* Puede ser difícil encontrar soporte técnico.

Algunas librerías de “*OPEN SOURCE*” reconocidas son JQUERY, REACT.JS, MOMENT.JS

***¿Que son librerías “CLOSED SOURCE”?***

“CLOSED SOURCE” se significa que el público no tiene acceso al código fuente. El publico no lo puede bajar, ni ver, ni usar esto al menos de que page, se suscriba o cree un plan de pagos y de esta forma tendrá acceso a ello.

Ventajas:

* Mas probable que sea un producto estable y enfocado.
* servicio al cliente suele ser más fácil de acceder.

Desventajas:

* **A menudo cuesta dinero.**
* **Tener que esperar al creador si te faltan funciones o hay errores en librería.**

**Ejemplo: USO DE LIBRERÍA OPEN SOURCE “MOMENT.JS”**

1. Vamos a la página de Moment.js y descargamos archivo

Captura de pantalla de computadora

Descripción generada automáticamenteInterfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

1. Lo agregamos a nuestro Proyecto.  
   -- Agregamos una carpeta donde vamos a almacenar todos nuestros objetos .JS Texto

   Descripción generada automáticamente  
   -- Le damos agregar elemento existente   
   Interfaz de usuario gráfica, Texto, Aplicación

   Descripción generada automáticamente  
     
   -- Seleccionamos el objeto y lo agregamos a carpeta “js”  
   Interfaz de usuario gráfica, Texto, Aplicación

   Descripción generada automáticamenteInterfaz de usuario gráfica, Texto

   Descripción generada automáticamente
2. Referenciamos en el objeto donde queremos utilizar la librería.  
   --VERDE: La referencia al JS se hace con un tag de script con tipo=”text/javascript” y le indicamos la ruta de nuestro objeto.  
   Interfaz de usuario gráfica, Texto, Aplicación

   Descripción generada automáticamente
3. Agregamos validación sobre campo fecha de nacimiento utilizando la librería de “*momento.js*”.  
   --Creamos función “addFecha” y se la pasamos al campo de fecha de nacimiento.  
   Interfaz de usuario gráfica, Texto, Aplicación

   Descripción generada automáticamente  
     
     
   --Resultado: El campo Fecha de nacimiento solo acepta fecha con formato “AAAA-MM-DD”.  
   Interfaz de usuario gráfica

   Descripción generada automáticamente

## 2.- Framework

**¿Qué es un “FRAMEWORK”?**

Framework, por otro lado, es como la base sobre la cual los desarrolladores construyen aplicaciones para plataformas específicas.

Los marcos son como bibliotecas en el sentido de que facilitan su trabajo, pero no puede llamar a los marcos de la misma manera que a las bibliotecas. Si desea usar el marco, debe aprender el marco, porque en este caso, el marco llama a su código, y no al revés.

Un desarrollador no llama a un marco; en cambio, es el marco es el que llamará y usará el código de alguna manera particular.

Diagrama

Descripción generada automáticamente

Framework te llama

TU llamas la libreria

Framework contiene librerias

Tu codigo

Algunos Frameworks reconocidos son Angular, Bootstrap, Laravel y Vue.

**Ejemplo**: Uso de Framework Bootstrap.

1. Vamos a la página de Bootstrap y descargamos Bootstrap.  
   Interfaz de usuario gráfica, Aplicación, Teams

   Descripción generada automáticamente Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

   Descripción generada automáticamente
2. Agregamos los objetos de Bootstrap a nuestro proyecto.  
   --Descomprimimos y vemos que tenemos “JS” y “CSS”  
   Interfaz de usuario gráfica, Texto, Aplicación

   Descripción generada automáticamente  
   --Creamos una carpeta con el nombre “CSS” y le damos click en agregar objeto existente.  
   Interfaz de usuario gráfica, Texto, Aplicación

   Descripción generada automáticamente  
   --Seleccionamos el objeto y tipo CSS va en carpeta “CSS” y tipo JS en carpeta “JS”.  
   Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

   Descripción generada automáticamente Interfaz de usuario gráfica, Texto

   Descripción generada automáticamente  
     
   --La estructura de nuestras carpetes deberia de quedar de la siguiente manera.  
   Interfaz de usuario gráfica, Texto, Aplicación

   Descripción generada automáticamente
3. Referenciamos Bootstrap en el objeto donde vamos a implementar.  
   --VERDE: La referencia al CSS se hace con un tag de Style con tipo=”text/css” y le indicamos la ruta de nuestro objeto.  
   --MORADO: La referencia al JS se hace con un tag de script con tipo=”text/javascript” y le indicamos la ruta de nuestro objeto.Texto

   Descripción generada automáticamente con confianza baja
4. Hacemos una pequeña prueba con sintaxis de Bootstrap.  
   --Ponemos fondo negro con “bg-black” y texto blanco con “text-light” sobre nuestro <h3>CLIENTES:</h3>.  
   Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

   Descripción generada automáticamente  
   --Vemos resultado  
   Interfaz de usuario gráfica

   Descripción generada automáticamente  
   Listo ya le podemos dar formato y agregar estilos a nuestra vista con el framework “Bootstrap”.

## 3.- Paquetes Nuget

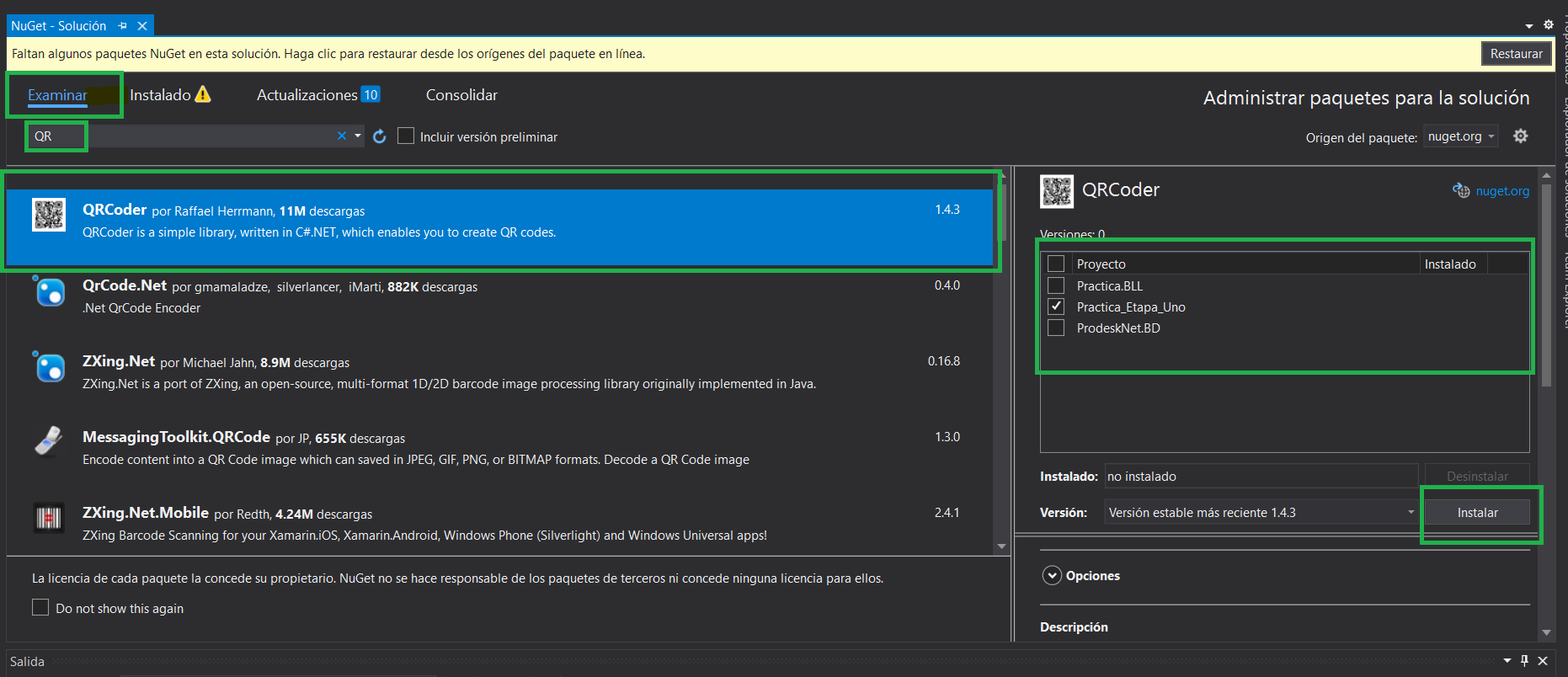
**¿Qué es un paquete Nuget?**

 Un paquete NuGet es una librería(DLL) que contiene código compilado en un único archivo ZIP con la extensión .nupkg y otros archivos relacionados con ese código y un manifiesto descriptivo que incluye información como el número de versión del paquete.

**Ejemplo**: Instalación y uso de Paquete Nuget

--Tenemos un manejador de paquetes nugets, dependiendo de la versión de Visual Studio el punto de acceso es diferente. Pero siempre buscamos “Nuget Package Manager” allí veremos todos los paquetes.

1. Vamos a “Nuget Package Manager”.  
   Interfaz de usuario gráfica, Texto, Aplicación

   Descripción generada automáticamente
2. Seleccionamos paquete que queremos instalar.  
   --Nos vamos a la pestaña examinar. Seleccionamos el paquete y el proyecto donde lo vamos a usar y damos click en instalar  
     
   --En la salida nos sale el status de la instalación de paquete.  
   Texto

   Descripción generada automáticamente
3. Implementación  
   --Importamos librería a nuestro objeto donde vamos a utilizar la librería para vb es con la palabra “imports” y C# con la palabra “using”  
     
   Captura de pantalla de un celular

   Descripción generada automáticamente  
     
   --Utilizamos los métodos y código definido de librería para generar nuestro resultado.  
   Texto

   Descripción generada automáticamente

## Documentación

* [librerías](https://dev.to/lmolivera/library-vs-framework-3bgo)
* [Open Source VS Closed Source](https://edu.gcfglobal.org/en/basic-computer-skills/open-source-vs-closed-source-software/1/)
* [Frameworks and libraries](https://www.interviewbit.com/blog/framework-vs-library/#:~:text=Definition,build%20applications%20for%20specific%20platforms.)
* [Paquete Nuget](https://docs.microsoft.com/en-us/nuget/what-is-nuget)