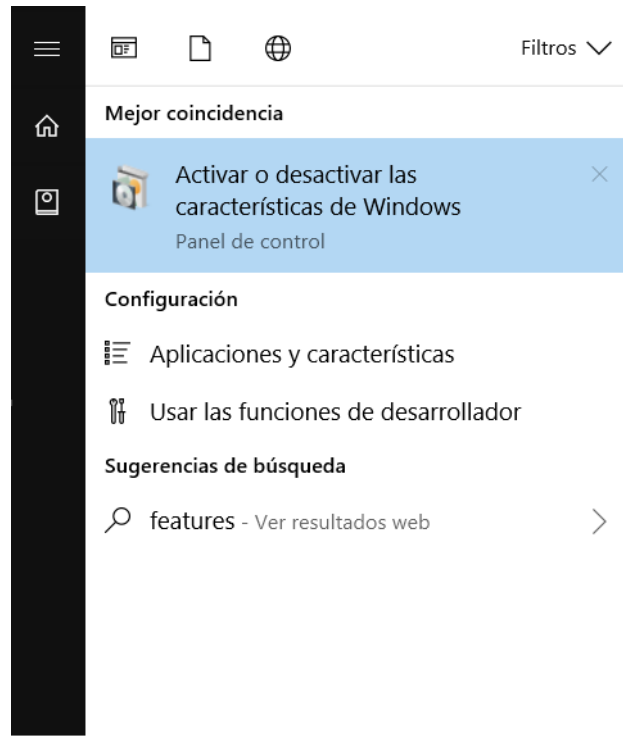
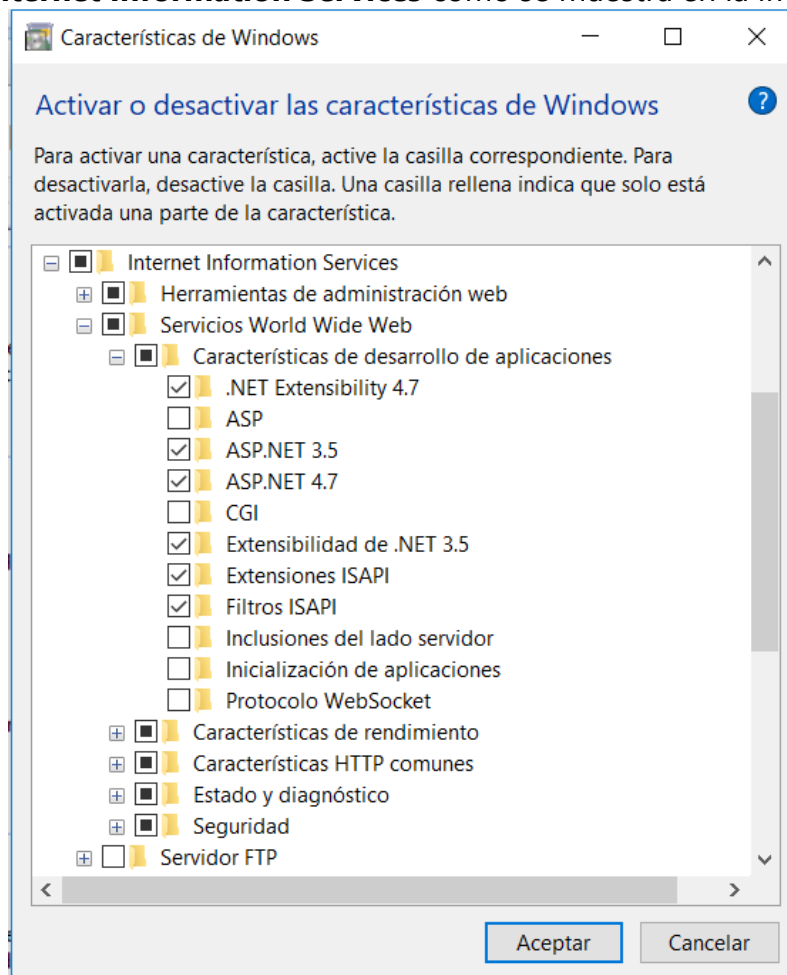


# 1. CONFIGURACIÓN DEL IIS

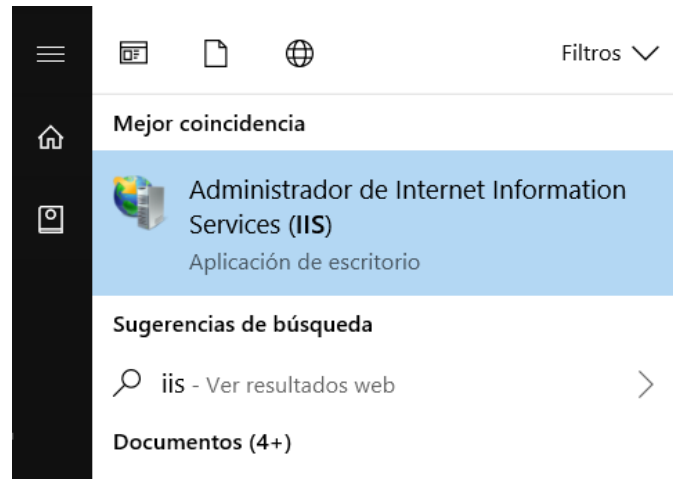
1.1) Se accede a las características de Windows:



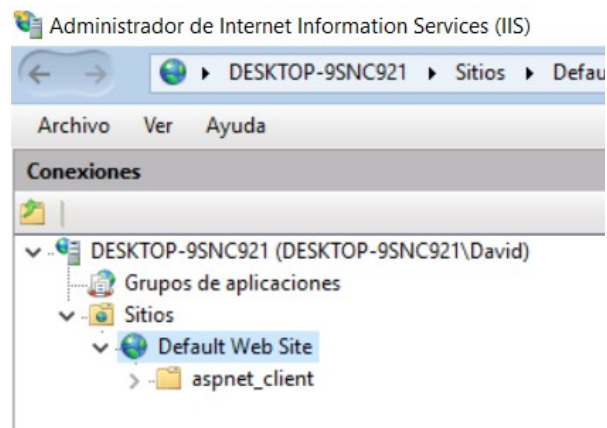
1.2) Configurar **Internet Information Services** como se muestra en la imagen:



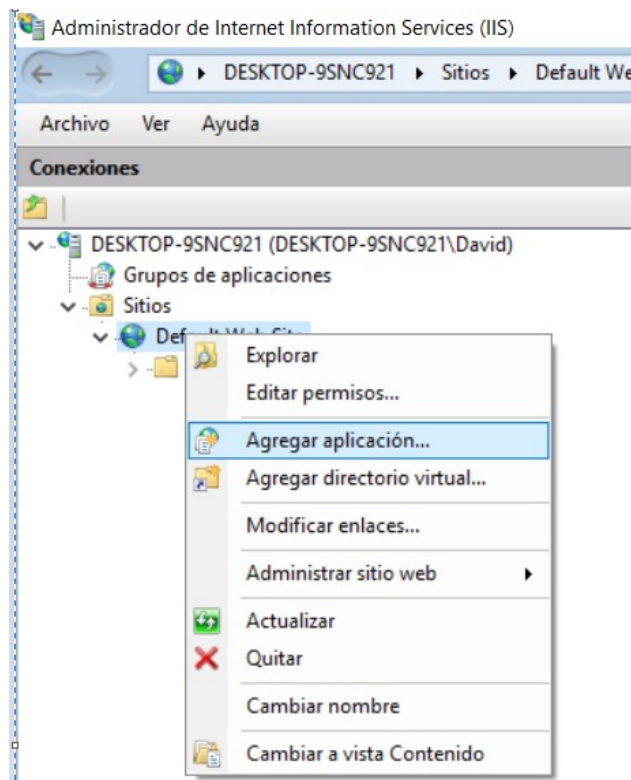
1.3) Acceder al administrador de **Internet Information Services**:



1.4) Deplegar el árbol hasta el Default Web Site:



1.5) Pulsar con el botón derecho sobre **Default Web Site** y pulsar sobre **Agregar Aplicación**:



1.6) En la ventana que aparece, rellenar **Alias** y **Ruta de acceso física**:

Agregar aplicación

Nombre del sitio: Default Web Site  
Ruta de acceso: /

Alias: SaladilloFitVR      Grupo de aplicaciones: DefaultAppPool      Seleccionar...

Ejemplo: ventas

Ruta de acceso física: C:\PRO\SaladilloFitVR      ...

Autenticación de paso a través  
Conectar como...      Probar configuración...

☐ Activar carga previa

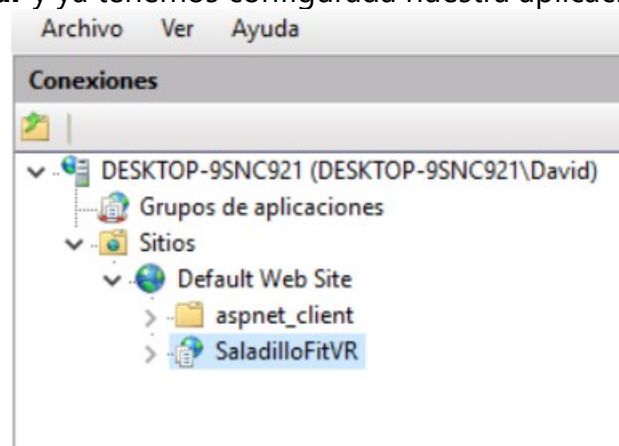
Aceptar      Cancelar

**Alias:** Nombre de nuestra aplicación. Se necesita acceder a los servicios que publique y es lo que se pone tras la dirección IP:

Por ejemplo, si el ordenador tiene la dirección IP 192.168.1.5 y en Alias hemos puesto SaladilloFitVR, las URL para acceder deberían comenzar por <http://192.168.1.5/SaladilloFitVR/>

**Ruta de acceso física:** Es la carpeta de nuestro ordenador donde se encuentra publicada la aplicación.

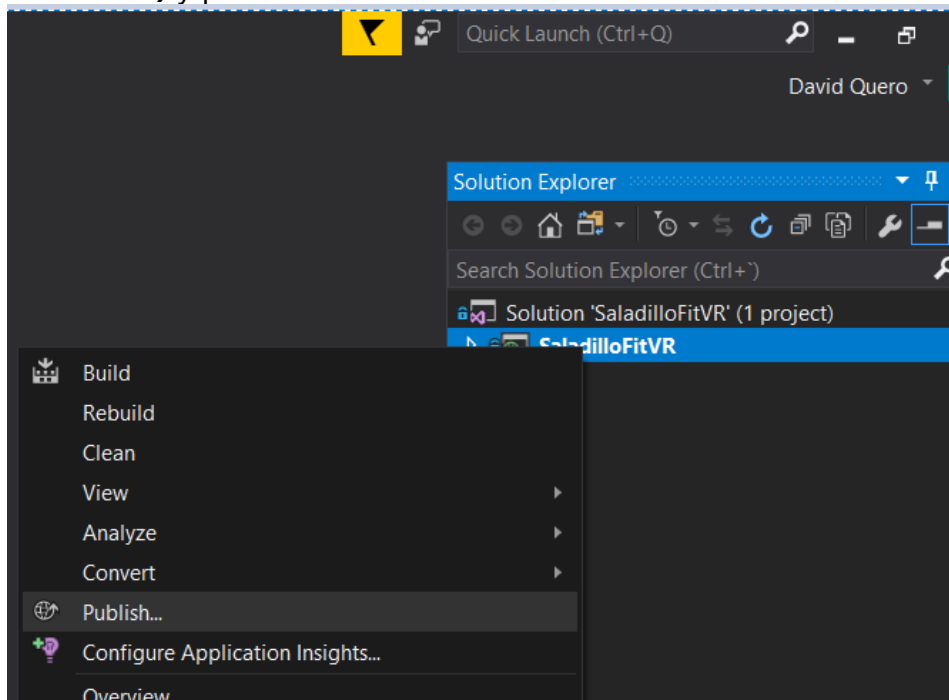
Damos a **Aceptar** y ya tenemos configurada nuestra aplicación:



## 2. PUBLICACIÓN DE LA WEB API

2.1) Se abre la Web API en Visual Studio 2017.

2.2) Se pulsa con el botón derecho sobre el proyecto (**OJO, SOBRE EL PROYECTO, NO SOBRE LA SOLUTION**) y pulsamos sobre **Publish**.



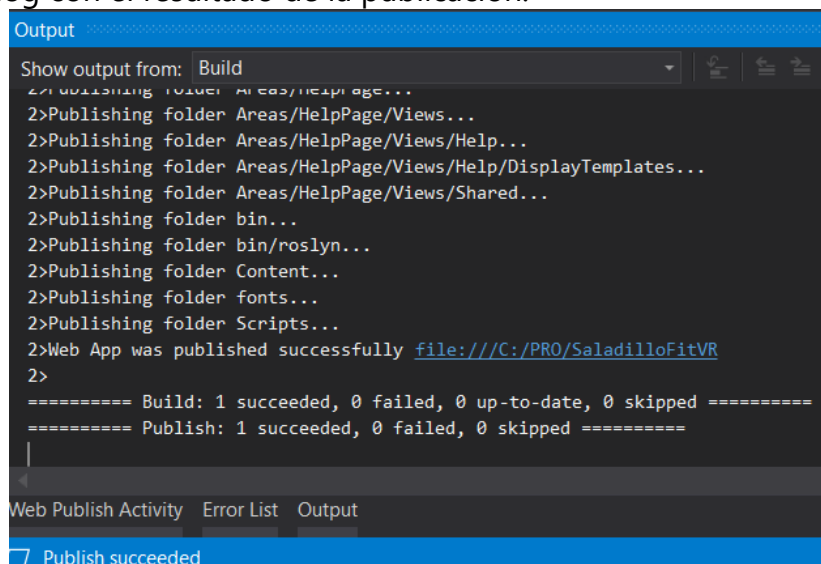
2.3) En la ventana que aparece, realizamos las siguientes acciones en el orden que se indica a continuación para publicar nuestra aplicación:

2.3.1) Seleccionamos **Folder**

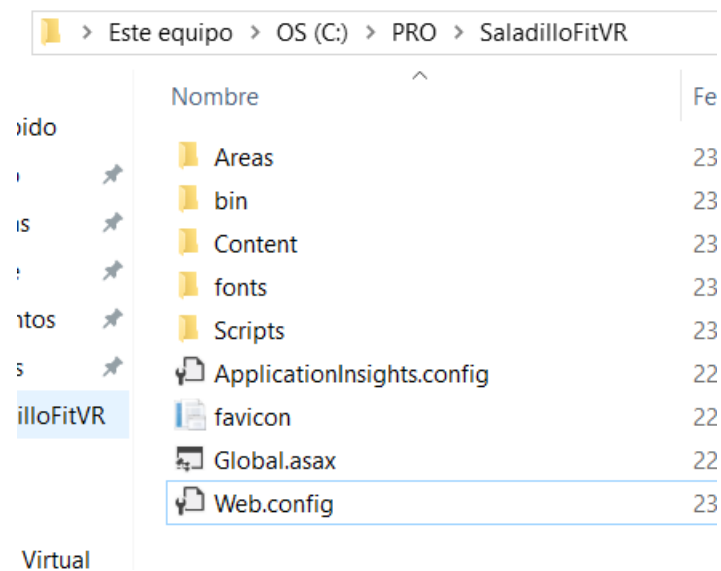
2.3.2) Con **Browse** seleccionamos la carpeta donde se va a publicar la aplicación, que debe coincidir con la **Ruta de acceso física** indicada en el **IIS**.

2.3.3) Damos al botón **Publish**.

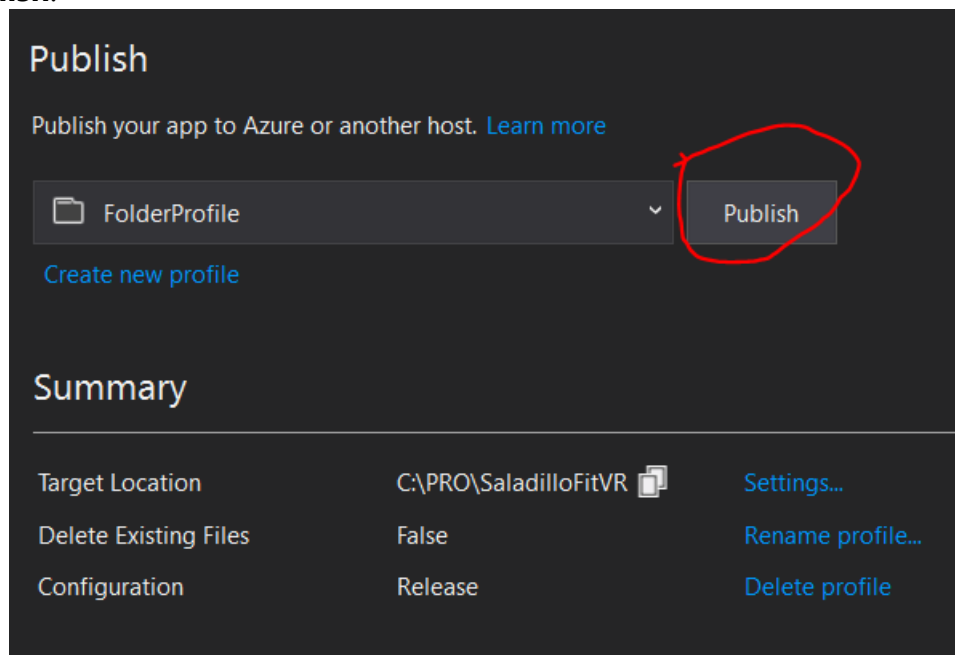
2.4) Aparece el log con el resultado de la publicación:



2.5) Si vamos a la carpeta indicada, vemos cómo se ha generado la publicación:

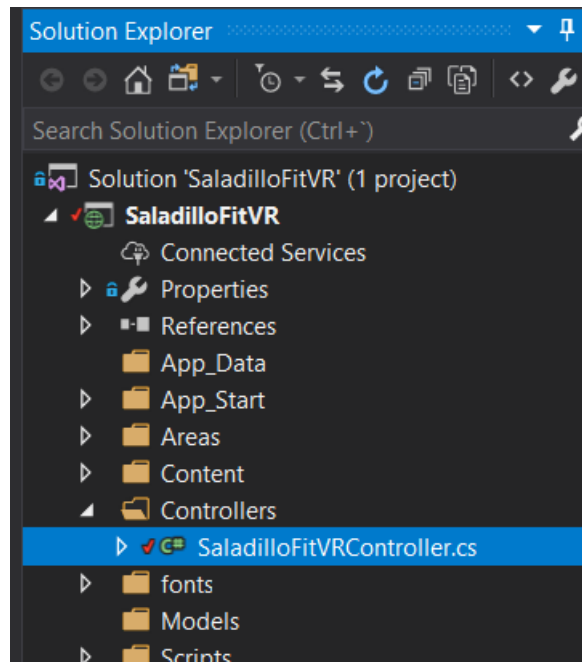


2.6) Para sucesivas publicaciones, ya está todo configurado, sólo será necesario darle al botón **Publish**:



### 3. CAMBIOS EN LA WEB API

3.1) Se abre el Controller **SaladilloFitVRController**:



3.2) El Controller está dividido entre métodos públicos que son los que exponen los servicios, y métodos privados que realizan la lógica necesaria y simulan la llamada a la base de datos. Ahí es donde tenéis que hacer los cambios para la simulación:

```
/// <summary>
/// Obtiene el nombre del cliente para el DNI recibido como parámetro.
/// </summary>
/// <remarks>
/// - Llama al método privado GetClientName.
/// - Converts the entity returned by the previous call to a models using the ModelConversion method from the ModelsHelper.
/// - Serializes the model to be returned by the method.
/// </remarks>
/// <param name="dni">DNI del cliente.</param>
/// <returns>
/// HttpResponseMessage with the result of the execution:
/// - StatusCode = code of the exception
/// - Content: data to be returned
/// </returns>
[HttpGet]
public HttpResponseMessage GetClientName(string dni)
{
    // Create the response
    var response = this.Request.CreateResponse(HttpStatusCode.OK);

    // Call the GetCmsInfo method and convert and serialize the data returned
    response.Content = new StringContent(JsonConvert.SerializeObject(GetClientNameInfo(dni)), Encoding.UTF8, "application/json");

    // Return the response
    return response;
}

#endregion

#region Métodos Privados

/// <summary>
/// Obtiene el nombre del cliente para el DNI recibido como parámetro.
/// </summary>
/// <param name="dni">DNI del cliente.</param>
/// <returns>Nombre del cliente.</returns>
private string GetClientNameInfo(string dni)
{
    // Nombre del cliente
    string name = string.Empty;

    //TODO: Añadir el código para pruebas

    // Devuelve el nombre del cliente
    return name;
}
```