

Nº45 septiembre 2020



Comparti MOSS

REVISTA ESPECIALIZADA EN TECNOLOGÍAS MICROSOFT

Entrevista
Mayra Badillo
Villamizar

Jupyter,
Azure ML y
Databricks

Integrando Azure
Cognitive Ser-
vices con Share-
Point Online

Introducción a
XMLA Endpoints
en Power BI

Staff

CompartiMOSS es una publicación independiente de distribución libre en forma electrónica. Las opiniones aquí expresadas son de estricto orden personal, cada autor es completamente responsable de su propio contenido.

DIRECCIÓN GENERAL

- Gustavo Velez
- Juan Carlos Gonzalez
- Fabian Imaz
- Alberto Diaz

DISEÑO Y DIAGRAMACIÓN

- Santiago Porras Rodríguez

Contacte con nosotros

revista@compartimoss.com
gustavo@gavd.net
jcgonzalezmartin1978@hotmail.com
fabian@siderys.com.uy
adiazcan@hotmail.com

BLOGS

<http://www.gavd.net>
<https://jcgonzalezmartin.wordpress.com/>
<http://blog.siderys.com>
<https://adiazcan.github.com>

REDES SOCIALES

Facebook:

<http://www.facebook.com/group.php?gid=128911147140492>

LinkedIn:

<http://www.linkedin.com/groups/CompartiMOSS-3776291>

Twitter:

@CompartiMOSScom

Contenido

03

Editorial

07

Asp.Net Core - Gestionando Cookies De Autenticacion Server Side

13

Azure Static Web Apps: ¿Qué es este nuevo servicio?

20

Gestión de servidores externos mediante Azure Policy

30

Jupyter, Azure ML y Databricks

35

Entrevista Kabel

42

Powerapps, Azure Maps y Geoespatial Features

50

ASP.Net Core con ReactJS

04

Usando el Detector de Anomalías de Azure con Office 365

10

Consumiendo el nuevo endpoint TO DO de MS Graph API desde un SPFx Outlook Add-in con SharePoint Online

17

Entrevista Mayra Badillo Villamizar

23

Un repaso a las herramientas de Microsoft para migrar a SharePoint Online y OneDrive For Business

32

Asignación de políticas en Microsoft Teams

38

Integrando Azure Cognitives Services con SharePoint Online

48

Introducción a Azure SQL

52

Introducción a XMLA Endpoints en Power BI



03

Editorial

La primera mitad del año 2020 comenzó con buenas perspectivas, y termino dándonos motivos de preocupación. La segunda mitad sigue presentándonos tiempos difíciles a todo nivel: para las personas individualmente, las empresas y las economías en todo el mundo. La estabilidad está muy lejos de ser alcanzada, sea que hablemos de estabilidad política en un mundo cada vez más polarizado, económica cargada con incertidumbres a montón y certezas que no existen, y, de nuevo, personal debido a las otras dos, y por todo lo que hemos tenido que asimilar en cuanto a (deprivación de) relaciones interpersonales, y nuevas formas de interactuar con nuestros amigos, conocidos y colegas de trabajo.

Satya Nadella, el CEO de Microsoft, dijo recientemente que "hemos visto dos años de transformación digital en dos meses" muy acertadamente. Aunque en nuestro mundo de computadores e informática estamos acostumbrados desde hace años a trabajar remotamente, de todas formas y por primera vez estamos experimentando que es hacer proyectos sin ninguna manera de contacto personal, todo realizado virtualmente. Y, aunque para nosotros puede ser fácil adaptarnos a esta nueva realidad, para muchos otros sectores de nuestra sociedad ese cambio puede ser traumático. Y aquí es donde nace nuestra responsabilidad, no solo técnica sino también humana, y casi, casi ética, de hacer que nuevas tecnologías sean enseñadas y aprendidas de forma sencilla por no-expertos, que todos nuestros avances técnicos sean estables, usables, entendibles.

Aunque parezca raro, nosotros, como tecnócratas en un mundo en crisis, tenemos una responsabilidad social muy grande. En cierta forma, el superar el universo en crisis que estamos viviendo en este momento depende de nuestra capacidad para asimilar nuevas tecnologías, evitar que la tecnología se convierta en nuestra enemiga, o que sea utilizada para destruirnos, separarnos, profundizar la brecha, en lugar de servir como la goma que une todos nuestros esfuerzos comunitarios para crear un mundo mejor.

CompartiMOSS nació hace ya casi quince años con un elemento altruista por excelencia: distribuir información técnica sin ningún ánimo de lucro, solamente porque creemos en que todos podemos ser mejores si intercambiamos nuestro conocimiento. En esta época de crisis seguimos combatiendo por los mismos motivos. Los editores de la revista, junto con sus cientos de autores y miles de lectores, estamos luchando constantemente para conseguir ese objetivo. Esperamos poder seguir haciéndolo los próximos quince años también, y que la revista no se convierta en una "víctima colateral" de la crisis actual.

Disfruten este nuevo número de CompartiMOSS tanto como nosotros, los editores y autores, disfrutamos creándolo.

El Equipo Editorial de CompartiMOSS



Usando el Detector de Anomalías de Azure con Office 365

El Anomaly Detector es un servicio de Azure que permite monitorear series de datos en el tiempo para detectar incoherencias utilizando Inteligencia Artificial. El Detector aplica automáticamente algoritmos de análisis de datos, independientemente de su tipo, escenario o volumen. Con los datos de series distribuidos en el tiempo, el Detector determina los límites para la localización de anomalías, los valores esperados y qué puntos se apartan de lo esperado. El uso del Anomaly Detector no requiere ningún conocimiento especial o experiencia con Inteligencia Artificial y dispone de dos APIs, uno utilizando REST y el otro usando clases nativas de DotNet, que permiten manejarlo programáticamente.

"El Anomaly Detector es un servicio de Azure que permite monitorear series de datos en el tiempo para detectar incoherencias"

El servicio se puede utilizar en dos formas:

- Analizando datos en batch. Usa una serie temporal de datos para detectar cualquier anomalía que pueda existir a lo largo de la serie. Esta operación genera un modelo utilizando todos los datos de la serie, y analiza cada punto con el modelo.
- Analizando datos en tiempo-real. Detecta anomalías en los datos enviados en tiempo real utilizando puntos de datos enviados anteriormente para determinar si el último es una anomalía. Esto permite analizar datos a medida que se van creando.

Hay varios escenarios en donde el Detector de Anomalías se puede utilizar en Office 365, el más aparente de ellos es para detectar cambios en el uso de los servicios de Office. Microsoft 365 ofrece estadísticas de utilización de la mayoría de sus servicios, tales como el uso de SharePoint, Teams, Exchange o OneDrive For Business. Cuando uno de estos servicios recibe un ataque externo, o hay un crecimiento abrupto por algún tipo de evento, el Detector de Anomalías puede señalar los cambios y avisar a los administradores del sistema.

Por ejemplo, el registro de páginas únicas visitadas en SharePoint por 90 días en un tenant de Office 365 puede aparecer como el que muestra la siguiente imagen.

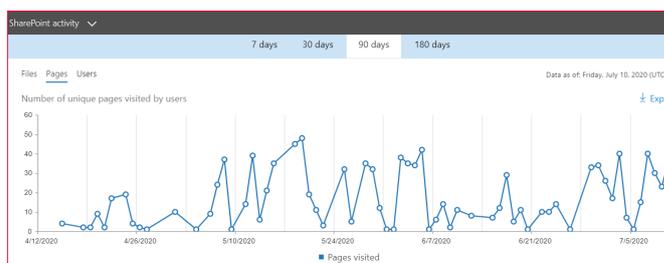


Imagen 1.- Serie de datos en el tiempo que va a ser analizada

Note que la serie se puede descargar en formato .csv utilizando el botón de "Export" en la esquina superior derecha.

El archivo .csv se le puede ofrecer al Anomaly Detector para que lo analice, e indique cuales puntos merecen atención. El resultado en una aplicación de consola sería como indica la siguiente imagen:

```
C:\Projects\CompartimossAnomalyDetector\bin\Debug\CompartimossAnomalyDetector.exe
Detecting anomalies in the entire time series.
An anomaly was detected at index:
29 37 40 41 47 50 51 55 56 57 70 71 72 74 75 76 77 82 83 84 85 86 89
```

Imagen 2.- Resultados del análisis.

Indicando que el pico de bajada en los días 70 a 77 y 82 a 86 se debe analizar con cuidado.

Creación del servicio en Azure

Para utilizar el Detector de Anomalías es necesario crear primero el servicio en Azure. Para acceder a la página de administración de Azure (<https://portal.azure.com>) necesita credenciales con suficientes derechos (es posible obtener una cuenta temporal de prueba de Azure desde la página de Microsoft <https://azure.microsoft.com/en-us/try/cognitive-services/>).

Para crear una instancia del servicio en Azure:

- 1.- Entre al portal de manejo de Azure (<https://portal.azure.com>) utilizando sus credenciales.
- 2.- Vaya a la sección de "Resource Groups" y cree un nuevo Grupo de Recursos (también es posible reutilizar un grupo ya existente).
- 3.- Cree un servicio de "Computer Vision API":
 - En el Resource Group, utilice el botón de "+Add" para crear un recurso, busque por "anomaly detector" en la casilla de búsqueda y seleccione "Ano-

malý Detector” en los resultados. En el momento de escribir este artículo, el servicio se encontraba todavía en “Preview”.

- Asígnale un nombre al servicio y utilice el Grupo de Recursos deseado. En la casilla de “Pricing tier” seleccione un nivel dependiendo de la cantidad de consultas a esperar por segundo, lo que determina el precio del servicio (por llamadas por segundo). El nivel “F0” es gratis y ofrece un máximo de 10 llamadas por segundo y 20.000 transacciones por mes.

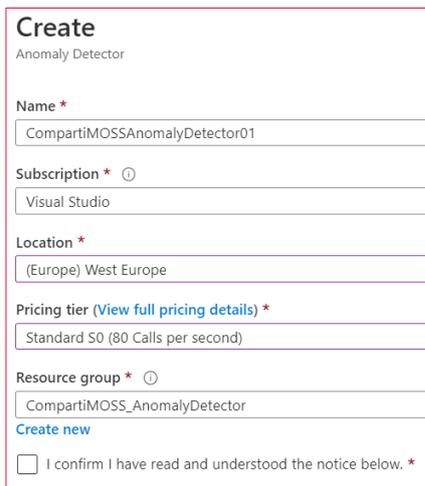


Imagen 3.- Creación del servicio de Anomaly Detector.

- 4.- Una vez creado el servicio, haga clic sobre su nombre en la lista de recursos del Resource Group, vaya a “Keys and Endpoints” y copie el valor de “Key 1” y “Endpoint”.



Imagen 4.- Ventana con los valores de las llaves y punto de entrada.

Usando el Detector de Anomalías

Azure ofrece dos API para trabajar con el Detector de anomalías: un REST API, y una librería de clientes para C#, JavaScript y Python. El siguiente ejemplo utiliza una aplicación de consola de .NET Framework programada en C#.

- 5.- Después de crear la solución en Visual Studio, instale el paquete de NuGet llamado “Microsoft.Azure.CognitiveServices.AnomalyDetector”. El paquete instala todas las referencias necesarias. El código para el método Main es como se indica a continuación.

```
static void Main(string[] args)
{
    string myEndpoint = "https://compartimossanomalydetec-
```

```
tor.cognitiveservices.azure.com/";
    string myKey = "bf7c186c66484cce9c676aacc966af77";

    string csvFilePath = @"C:\Projects\CompartimossAnomaly-
    Detector\" +
        "SharePointSiteUsagePages7_12_2020 12_59_55
    PM.csv";

    IAnomalyDetectorClient myClient = new AnomalyDetector-
    Client(new
        ApiKeyServiceClientCredentials(myKey))
    {
        Endpoint = myEndpoint
    };

    Request myRequest = GetPointsFromCsv(csvFilePath);

    EntireDetectSample(myClient, myRequest);
    LastDetectSample(myClient, myRequest);

    Console.WriteLine("InPress ENTER to exit");
    Console.ReadLine();
}
```

Configure en las dos primeras variables los valores del servicio de Azure obtenidos en el punto 4. La tercera variable indica la localización del archivo .csv con la serie de datos históricos a analizar, en este caso los datos de uso de páginas en sitios de SharePoint obtenido exportando el reporte respectivo desde la administración central de Office 365 (Imagen 1).

- 6.- Utilizando el punto de entrada del servicio de Azure se crea un cliente de acceso utilizando la clase IAnomalyDetectorClient. La rutina GetPointsFromCsv convierte los valores del archivo .csv en un objeto del tipo “Request”.

La rutina GetPointsFromCsv utiliza una librería para leer archivos .csv llamada “CsvHelper” (<https://josh-close.github.io/CsvHelper/>), que se puede instalar en la solución de Visual Studio por medio del paquete de NuGet del mismo nombre. El código de la rutina es:

```
static Request GetPointsFromCsv(string csvPath)
{
    List<Point> myPointsList = new List<Point>();

    using (StreamReader myReader = new StreamReader(csv-
    Path, Encoding.UTF8))
    using (CsvReader myCsv = new CsvReader(myReader, Cultu-
    reInfo.InvariantCulture))
    {
        myCsv.Read();
        myCsv.ReadHeader();
        while (myCsv.Read())
        {
            Point onePoint = new Point
            {
                Timestamp = myCsv.GetField<DateTime>("Report Date"),
                Value = myCsv.GetField<Double>("Page View Count")
            };
            myPointsList.Add(onePoint);
        }
    }

    // List MUST be ordered by date
    List<Point> myPointsListOrdered = myPointsList.OrderBy(o
    => o.Timestamp).ToList();

    return new Request(myPointsListOrdered, Granularity.Daily);
}
```

El archivo .csv exportado por Office 365 contiene las siguientes columnas:

	1	2	3	4	5	6	7
1	Report Refresh Date	Site Type	Page View Count	Report Date	Report Period		
2	2020-07-06	All	119	2020-07-06	90		
3	2020-07-06	All	2	2020-07-05	90		
4	2020-07-06	All	9	2020-07-04	90		
5	2020-07-06	All	172	2020-07-03	90		
6	2020-07-06	All	67	2020-07-02	90		

Imagen 5.- Formato del archivo csv exportado desde Office 365.

La definición de la clase “Request” requiere una Lista genérica con valores del tipo “Point” que contiene solo dos valores: la fecha de cada punto, y su valor numérico. Además, la clase exige que los valores deben ser organizados de mas antiguo a más reciente, mientras que los valores exportados desde Office tienen el orden contrario. El servicio de Detector de Anomalías de Azure requiere un mínimo de 12 puntos para analizar, y acepta un máximo de 8.640 puntos.

"El servicio utiliza Inteligencia Artificial para crear sus propios modelos de análisis"

La rutina GetPointsFromCsv primero carga el archivo en un StreamReader, y luego extrae solamente los valores de las columnas “Report Date” y “Page View Count” para crear la Lista genérica. Después de leer todos los valores del archivo csv, el orden de los valores se invierte utilizando una expresión Linq para que queden en el orden requerido por el servicio. El valor del objeto “Request” requiere también que se indique el tipo de valores de la serie, y en este caso se indica que los valores son diarios por medio de la enumeración “Granularity”.

7.- La rutina “EntireDetectSample” analiza la serie de datos completos en un Batch, y retorna todos los valores sospechosos. La rutina es de la forma:

```
static void EntireDetectSample(IAnomalyDetectorClient AnomalyClient, Request AnomalyRequest)
{
    Console.WriteLine("Detecting anomalies in the entire time series");

    EntireDetectResponse result = AnomalyClient.EntireDetectAsync(AnomalyRequest).Result;

    if (result.IsAnomaly.Contains(true))
    {
        Console.WriteLine("An anomaly was detected at index:");
        for (int idx = 0; idx < AnomalyRequest.Series.Count; ++idx)
        {
            if (result.IsAnomaly[idx])
            {
                Console.Write(idx);
                Console.Write(" ");
            }
        }
        Console.WriteLine();
    }
    else
    {
        Console.WriteLine(" No anomalies detected in the series");
    }
}
```

El método “EntireDetectAsync” del cliente se encarga de enviar los datos al servicio de Azure y retornar los puntos

sospechosos. Finalmente, la aplicación de consola renderiza los valores retornados.

8.- La rutina “LastDetectSample” es muy similar a la rutina anterior, pero utiliza el método “LastDetectAsync” para indicar si el ultimo valor de la serie histórica es sospechoso o no:

```
static void LastDetectSample(IAnomalyDetectorClient AnomalyClient, Request AnomalyRequest)
{
    Console.WriteLine("Detecting the anomaly status of the latest point in the series");
    LastDetectResponse result = AnomalyClient.LastDetectAsync(AnomalyRequest).Result;

    if (result.IsAnomaly)
    {
        Console.WriteLine("The latest point was detected as an anomaly");
    }
    else
    {
        Console.WriteLine("The latest point was not detected as an anomaly");
    }
}
```

9.- El resultado que presentan las dos rutinas de análisis combinadas es como indica la siguiente figura.

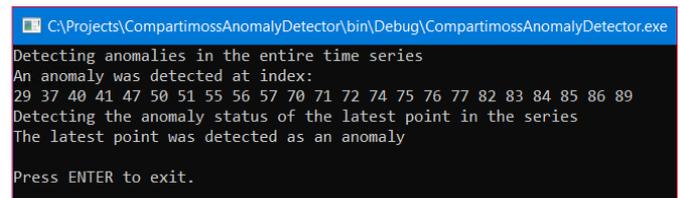


Imagen 6.- Resultado del análisis.

Conclusión

El Detector de Anomalías de Azure es un servicio que permite detectar valores que sospechosos en una serie de datos en el tiempo. El servicio utiliza Inteligencia Artificial para crear sus propios modelos de análisis, pero no requiere que los usuarios tengan ningún conocimiento sobre como crear estos modelos matemáticos. La implementación programática es muy sencilla, y dispone de dos APIs, uno por medio de REST y el otro utilizando clases nativas.

Referencias

<https://docs.microsoft.com/en-us/azure/cognitive-services/anomaly-detector/overview>

CsvHelper

<https://joshclose.github.io/CsvHelper/>

GUSTAVO VELEZ

MVP Office Apps and Services

gustavo@gavd.net

<http://www.gavd.net>

Asp.Net Core - Gestionando Cookies de Autenticación Server Side

En este artículo veremos cómo gestionar las cookies de autenticación de Asp.Net Core en el servidor. En el ejemplo veremos cómo usar una base de datos vía Entity Framework Core para persistir y gestionar las cookies. Os indico que este método es extensible para poder usar en vez de nuestro motor de base de datos favorito, un servicio de caché distribuida externo como Redis Caché.

"Asp.Net Core por defecto gestiona las cookies en cliente"

Asp.Net Core por defecto gestiona las cookies en cliente, pero si necesitamos una mayor seguridad que nos permita realizar un logout remoto, controlar el número de sesiones abiertas que tiene un usuario, inclusive eliminar problemas que pueden provocarse al serializar grandes datos en nuestras cookies de autenticación, es necesario el gestionar estas cookies en el servidor.

En el código de demo usaremos Identity Server para la gestión del login de los usuarios.

¿Cómo implementar esto?

Debemos crearnos una clase que implemente el interfaz ITicketStore cuyos métodos a implementar son los siguientes:

- Task RemoveAsync(string key); => Elimina la identidad asociada con la clave proporcionada.
- Task RenewAsync(string key, AuthenticationTicket ticket); => Actualiza la identidad cuya clave se ha proporcionado.
- Task<AuthenticationTicket> RetrieveAsync(string key); => Devuelve la identidad cuya clave se proporciona.
- Task<string> StoreAsync(AuthenticationTicket ticket); => Almacena el ticket de autenticación y devuelve una clave asociada.

Bien, una vez visto el interfaz que debemos implementar pasemos a algo de código. Lo primero es definir nuestra tabla de la base de datos

```
public class AuthenticationTicket
{
    public Guid Id { get; set; }
```

```
public string UserId { get; set; }
public byte[] Value { get; set; }
public DateTimeOffset? LastActivity { get; set; }
public DateTimeOffset? Expires { get; set; }
}
```

Como vemos, guardaremos el identificador del usuario, el contenido del ticket de autenticación, su tiempo de expiración para controlar la caducidad de la cookie y generaremos un identificador que es el que devolveremos al cliente y con el cual gestionaremos las operaciones de eliminación, renovación, devolución e inserción en nuestra base de datos. Una vez definido nuestra tabla o nuestro objeto de base de datos, procederemos a implementar un DbContext.

```
public class DataProtectionKeysContext : DbContext, IDataProtectionKeyContext
{
    public DataProtectionKeysContext(DbContextOptions<DataProtectionKeysContext> options)
        : base(options) {}
    public DbSet<DataProtectionKey> DataProtectionKeys { get; set; }
    public DbSet<AuthenticationTicket> AuthenticationTickets { get; set; }

    protected override void OnModelCreating(ModelBuilder modelBuilder)
    {
        modelBuilder.Entity<AuthenticationTicket>().ToTable("AuthenticationTicket").HasKey(t => new { t.Id });
        base.OnModelCreating(modelBuilder);
    }
}
```

Si os fijáis, en este DbContext, implemento un interfaz denominado IDataProtectionKeyContext para poder almacenar las claves usadas por Asp.Net Core DataProtection (Más adelante os explicaré en otro artículo sobre cómo configurar nuestras aplicaciones Asp.Net Core para desplegarlas en una granja de servidores) para cifrar/descifrar el contenido del ticket de autenticación).

En el método ConfigureServices de nuestra clase Startup tendrá el siguiente aspecto.

```
/// <summary>
```

```

/// ConfigureServices
/// </summary>
/// <param name="services"></param>
public void ConfigureServices(IServiceCollection services)
{
    services.AddDbContext<DataProtectionKeysContext>(options =>
        options.UseSqlite(
            Configuration.GetConnectionString("DataProtectionKeysConnection")));

    var encryptionSettings = new AuthenticatedEncryptorConfiguration()
    {
        EncryptionAlgorithm = EncryptionAlgorithm.AES_256_CBC,
        ValidationAlgorithm = ValidationAlgorithm.HMACSHA256
    };

    services.AddDataProtection()
        .PersistKeysToDbContext<DataProtectionKeysContext>()
        .SetApplicationName("demo")
        .UseCryptographicAlgorithms(encryptionSettings);

    services.AddAuthentication(options =>
    {
        options.DefaultScheme = "cookie";
        options.DefaultChallengeScheme = "oidc";
    })
        .AddCookie("cookie", options =>
    {
        options.ExpireTimeSpan = TimeSpan.FromMinutes(5);
        options.SlidingExpiration = true;
        options.SessionStore = new TicketStore(services, DataProtectionProvider.Create(this.GetType().FullName));
    })
        .AddOpenIdConnect("oidc", options =>
    {
        options.Authority = "https://localhost:5001";
        options.ClientId = "mvc.client";
        options.ClientSecret = "36F742BA-D9BF-49FE-B91A-D25E3A6354A5";

        // code flow + PKCE (PKCE is turned on by default)
        options.ResponseType = "code";
        options.UsePkce = true;

        options.Scope.Add("openid");
        options.Scope.Add("profile");
        options.Scope.Add("email");
        options.Scope.Add("scope1");
        options.Scope.Add("scope2");
        options.GetClaimsFromUserInfoEndpoint = true;
        options.SaveTokens = true;
    });

    services.AddControllersWithViews(options =>
    {
        var policy = new AuthorizationPolicyBuilder()
            .RequireAuthenticatedUser()
            .Build();
        options.Filters.Add(new AuthorizeFilter(policy));
    });
}

```

"Es sencillo implementar la gestión de cookies en servidor utilizando para ello el interfaz ITicketStore"

Procedo a explicar el código para que no os perdáis:

- Incluimos nuestro contexto de base de datos en el motor de inyección de dependencias.
- Agregamos los servicios de DataProtection e indicamos que las claves se deben incluir en el contexto de base de datos del punto anterior.

- Agregamos los servicios de autenticación en los cuales configuramos:

- El esquema por defecto de autenticación será el de Cookie de autenticación. Sobre la que configuramos tiempo de expiración y que use nuestro TicketStore para almacenar los tickets en nuestra base de datos, al que le pasamos el IServiceCollection para que dentro de nuestra clase podamos resolver fácilmente las dependencias necesarias y un DataProtectorProvider para realizar el cifrado/descifrado del ticket.

- Configurar OpenIdConnect estableciendo a nuestro Identity Server como proveedor de Autoridad.

- Agregamos los servicios para los controladores y vistas.

¿Cómo queda entonces la implementación de nuestro TicketStore?

La implementación quedaría de la siguiente forma:

```

/// <summary>
/// TicketStore
/// </summary>
public class TicketStore : ITicketStore
{
    private readonly IServiceCollection services;
    private readonly IDataProtector dataProtector;

    public TicketStore(IServiceCollection services, IDataProtectionProvider dataProtectionProvider)
    {
        this.services = services;
        this.dataProtector = dataProtectionProvider.CreateProtector(GetType().FullName);
    }

    /// <summary>
    /// RemoveAsync
    /// </summary>
    /// <param name="key"></param>
    /// <returns></returns>
    public async Task RemoveAsync(string key)
    {
        using var scope = services.BuildServiceProvider().CreateScope();
        var context = scope.ServiceProvider.GetService<DataProtectionKeysContext>();
        if (Guid.TryParse(key, out var id))
        {
            var authenticationTicket = await context.AuthenticationTickets.SingleOrDefaultAsync(x => x.Id == id);
            if (authenticationTicket != null)
            {
                context.AuthenticationTickets.Remove(authenticationTicket);
                await context.SaveChangesAsync();
            }
        }
    }

    /// <summary>
    /// RenewAsync
    /// </summary>
    /// <param name="key"></param>
    /// <param name="ticket"></param>
    /// <returns></returns>
    public async Task RenewAsync(string key, AuthenticationTicket ticket)
    {
        using var scope = services.BuildServiceProvider().CreateScope();
        var context = scope.ServiceProvider.GetService<DataProtectionKeysContext>();
        if (Guid.TryParse(key, out var id))
        {

```

```

    var authenticationTicket = await context.Authentication-
    Tickets.FindAsync(id);
    if (authenticationTicket != null)
    {
        authenticationTicket.Value = dataProtector.Protect(Se-
        rializeToBytes(ticket));
        authenticationTicket.LastActivity = DateTimeOffset.
       .UtcNow;
        authenticationTicket.Expires = ticket.Properties.Expi-
        resUtc;
        await context.SaveChangesAsync();
    }
}

/// <summary>
/// RetrieveAsync
/// </summary>
/// <param name="key"></param>
/// <returns></returns>
public async Task<AuthenticationTicket> RetrieveAsync(s-
tring key)
{
    using var scope = services.BuildServiceProvider().Crea-
    teScope();
    var context = scope.ServiceProvider.GetService<DataPro-
    tectionKeysContext>();
    if (Guid.TryParse(key, out var id))
    {
        var authenticationTicket = await context.Authentication-
        Tickets.FindAsync(id);
        if (authenticationTicket != null)
        {
            authenticationTicket.LastActivity = DateTimeOffset.
            UtcNow;
            await context.SaveChangesAsync();

            return DeserializeFromBytes(dataProtector.Unprotec-
            t(authenticationTicket.Value));
        }
    }

    return null;
}

/// <summary>
/// StoreAsync
/// </summary>
/// <param name="ticket"></param>
/// <returns></returns>
public async Task<string> StoreAsync(AuthenticationTic-
ket ticket)
{
    const string principalEmailType = "email";

    using var scope = services.BuildServiceProvider().Crea-
    teScope();
    var userId = ticket.Principal.FindFirst(t => t.Type == princi-
    palEmailType)?.Value;

```

```

    var context = scope.ServiceProvider.GetService<DataPro-
    tectionKeysContext>();

    var authenticationTicket = new AspNetCoreMvcClient.Data.
    AuthenticationTicket()
    {
        UserId = userId,
        LastActivity = DateTimeOffset.UtcNow,
        Value = dataProtector.Protect(SerializeToBytes(ticket)),
        Expires = ticket.Properties.ExpiresUtc
    };

    context.AuthenticationTickets.Add(authenticationTicket);
    await context.SaveChangesAsync();

    return authenticationTicket.Id.ToString();
}

private byte[] SerializeToBytes(AuthenticationTicket sour-
ce) => TicketSerializer.Default.Serialize(source);

private AuthenticationTicket Deserialize-
FromBytes(byte[] source) => source == null ? null : TicketSeri-
alizer.Default.Deserialize(source);
}

```

¿Dónde puedo encontrar el código de ejemplo?

Podéis descargaros los proyectos de ejemplo en: <https://github.com/sparraguerra/compartimoss/tree/master/AspNet-CoreManageCookieInServer>

Conclusiones

Hemos visto lo sencillo que es implementar la gestión de cookies en servidor utilizando para ello el interfaz ITicketStore. Podríamos ir más allá y poder almacenar por ejemplo desde qué IP ha iniciado sesión el usuario, con qué navegador o lo que se nos ocurra.

Happy coding!

SERGIO PARRA GUERRA

Software & Cloud Architect at Encamina

<https://twitter.com/sparraguerra>

i
10

Consumiendo el nuevo endpoint TO DO de MS Graph API desde un SPFx Outlook Add-in con SharePoint Online

Durante el pasado Build (Mayo 2020), Microsoft anunció un nuevo endpoint para su MS Graph API, que permite interactuar con Microsoft To Do, la app que pretende ser, en palabras del propio Microsoft: “el destino único para las tareas del usuario en Microsoft 365”

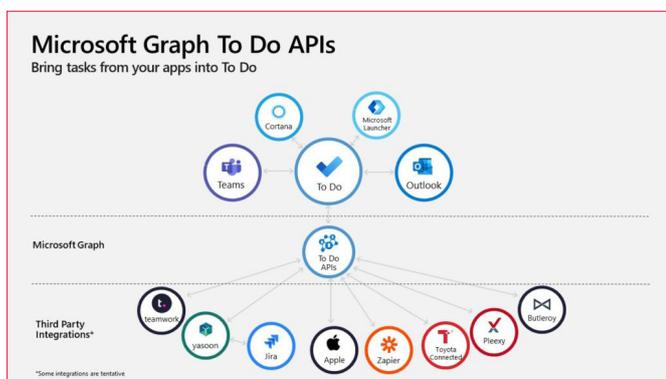


Imagen 1.- Microsoft Graph To Do APIs.

Dicho endpoint nos da la opción de crear y gestionar listas y tareas en To Do, así que es ideal para integraciones con sistemas externos, interfaces personalizadas, agregados, etc.

"la app que pretende ser, en palabras del propio Microsoft: el destino único para las tareas del usuario en Microsoft 365"

En este artículo vamos a ver cómo podemos consumir dicho endpoint desde un Web Part SPFx, que, además, vamos a desplegar también como una Add-in de Outlook (recuerda que un desarrollo SPFx puede desplegarse en SharePoint, Teams, o Outlook).

A día de hoy, el endpoint se encuentra en Preview, y todavía no está muy documentado, así que, antes de nada, vamos a ver qué operaciones tenemos disponibles, y como invocarlas usando Postman (<https://www.postman.com/>)

Invocando el To Do endpoint

Las tareas se crean dentro de Listas. La siguiente imagen describe las diferentes entidades que componen en endpoint:

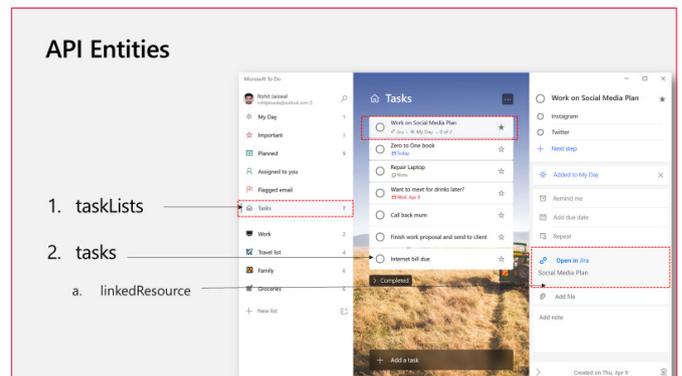


Imagen 2.- API Entities.

Para poder listar todas nuestras Listas, haremos un GET a la URL:

```
GET https://graph.microsoft.com/beta/me/todo/lists
```

La respuesta es un objeto JSON con un Array de entidades taskList:

```
{
  "@odata.context": "https://graph.microsoft.com/beta/$metadata#lists",
  "value": [
    {
      "@odata.etag": "W/\\\"yCVPuz60Uq3SCYMTc6PxxgAAyxwVw==\\\"",
      "displayName": "Flagged Emails",
      "isOwner": true,
      "isShared": false,
      "wellknownListName": "flaggedEmails",
      "id": "AAMkAGFjMjY0YzI1LTlTYNGQNGI1MC1iNTYzLWF1hZd1NmQ1MTZHNmAuAAAAAQAer1lWzrISanD6"
    },
    {
      "@odata.etag": "W/\\\"yCVPuz60Uq3SCYMTc6PxxgAA4ZBQEG==\\\"",
      "displayName": "Tasks",
      "isOwner": true,
      "isShared": false,
      "wellknownListName": "defaultList",
      "id": "AAMkAGFjMjY0YzI1LTlTYNGQNGI1MC1iNTYzLWF1hZd1NmQ1MTZHNmAuAAAAAQAer1lWzrISanD6"
    },
    {
      "@odata.etag": "W/\\\"yCVPuz60Uq3SCYMTc6PxxgAA4ZBQFA==\\\"",
      "displayName": "Community",
      "isOwner": true,
      "isShared": false,
      "wellknownListName": "none",
      "id": "AAMkAGFjMjY0YzI1LTlTYNGQNGI1MC1iNTYzLWF1hZd1NmQ1MTZHNmAuAAAAAQAer1lWzrISanD6"
    },
    {
      "@odata.etag": "W/\\\"yCVPuz60Uq3SCYMTc6PxxgAA4ZBQFG==\\\"",
      "displayName": "To Read Later",
      "isOwner": true,
      "isShared": false,
      "wellknownListName": "none",
      "id": "AAMkAGFjMjY0YzI1LTlTYNGQNGI1MC1iNTYzLWF1hZd1NmQ1MTZHNmAuAAAAAQAer1lWzrISanD6"
    }
  ]
}
```

"necesitas otorgar los permisos "Tasks.Read-Write" a la aplicación de SharePoint"

Podemos crear una nueva lista, si hacemos un POST de la


```
protected onInit(): Promise<void> {
  return new Promise((resolve, reject) => {
    this.context.msGraphClientFactory.getClient().then(client => {
      this._graphHttpClient = client;
      resolve();
    }).catch(error => {
      console.log(error);
      reject(error);
    });
  });
}
```

Lo siguiente que necesitamos es saber si estamos siendo ejecutados en el contexto de SharePoint, o de Outlook, ya que, en Outlook, tomaremos el título de la Task, del subject del email. El propio SPFx framework nos dará esa info del contexto, así que podemos hacer algo como:

```
private _getContext(): ICreateTaskContext {
  if (this.context.sdks.office) {
    const mailboxItem: Office.MessageRead = this.context.sdks.office.context.mailbox.item;
    const context: ICreateTaskContext = {
      graphHttpClient: this._graphHttpClient,
      siteUrl: this.context.pageContext.web.absoluteUrl,
      item: {
        id: mailboxItem.itemId,
        subject: mailboxItem.subject,
        from: mailboxItem.from.emailAddress,
        to: mailboxItem.to[0].emailAddress
      }
    };
    return context;
  } else {
    const context: ICreateTaskContext = {
      item: null,
      graphHttpClient: this._graphHttpClient,
      siteUrl: this.context.pageContext.web.absoluteUrl
    };
    return context;
  }
}
```

Como veis, primero comprobamos si tenemos contexto de Office, con la línea:

```
if (this.context.sdks.office) {
```

En ese caso, estamos en Outlook, y de nuevo el framework nos da opción de recuperar cierta información del email seleccionado, esto lo tenemos en la línea:

```
const mailboxItem: Office.MessageRead = this.context.sdks.office.context.mailbox.item;
```

Nuestro siguiente paso, ya dentro de un componente React, es obtener las diferentes Listas que tenemos en nuestro To Do App. El siguiente snippet se encarga de ello:

```
private async _getTodoLists(): Promise<ITodoList[]> {
  const endpoint: string = "https://graph.microsoft.com/beta/me/todo/lists";
  const response: any = await this.props.context.graphHttpClient
    .api(endpoint)
    .get();
  const graphResponse: any = response.value;

  const todoLists: ITodoList[] = graphResponse.map((item) => {
    const list: ITodoList = { id: item.id, displayName: item.displayName };
    return list;
  });

  return todoLists;
}

public async componentDidMount(): Promise<void> {
  const lists: ITodoList[] = await this._getTodoLists();
  this.setState({
    todoLists: lists,
  });
}
```

Una vez tenemos listadas nuestras listas, ya solo nos queda crear la nueva Task. Para ello podemos utilizar el siguiente snippet:

```
private async _createTask(): Promise<void> {
  try {
    if (!this.state.selectedList) {
      this.setState({
        showSelectListError: true
      });
      return;
    }

    const listId: string = this.state.selectedList.id;
    const taskTitle: string = this.state.newTaskTitle;

    const endpoint: string = `https://graph.microsoft.com/beta/me/todo/lists/${listId}/tasks`;

    const body: ITodoListItem = {
      importance: "high",
      status: "notStarted",
      title: taskTitle,
      body: {
        content: "You have a new task to do added from SPFx component!",
        contentType: "text",
      },
    };

    var response: any = await this.props.context.graphHttpClient
      .api(endpoint)
      .post(body);

    this.setState({
      listItemAdded: true
    });

    console.log(response);
  } catch (e) {
    console.log(e);
  }
}
```

Tenéis el proyecto completo como parte de los ejemplos de SPFx-Web Parts del PnP:

<https://github.com/pnp/sp-dev-fx-WebParts/tree/master/samples/react-outlook-add-todo-task>

Para su Deploy, se deben seguir los pasos bien documentados en los docs oficiales de Microsoft:

<https://docs.microsoft.com/en-us/sharepoint/dev/SPFx/office-addins-create#deployment-of-your-add-in>

Ten en cuenta que para utilizar el endpoint, necesitas otorgar los permisos "Tasks.ReadWrite" a la aplicación de SharePoint. Para ello te recomiendo que utilices el Office365 CLI (<https://pnp.github.io/office365-cli/>), con el siguiente comando:

```
spo serviceprincipal grant add --resource "Microsoft Graph"
--scope "Tasks.ReadWrite"
```

Espero haberos ayudado a entender como utilizar este nuevo endpoint.

¡Hasta el próximo artículo!

LUIS MAÑEZ
 Cloud Architect en ClearPeople LTD
 @luismanez
<https://github.com/luismanez>



Azure Static Web Apps: ¿Qué es este nuevo servicio?

Muchas veces cuando vamos a alojar una aplicación en Azure tenemos como principales servicios PaaS el uso de las Web App. Este servicio nos proporciona un recurso en algún servidor donde nosotros almacenamos nuestro código y se ejecuta con las especificaciones requeridas. Sin embargo, en algunas ocasiones estas necesidades son muy superiores a las necesidades que realmente se desean. En muchas veces solamente queremos mostrar contenido estático que se ejecuta en el FrontEnd y no tenemos ninguna necesidad de hacer ninguna llamada a nuestro BackEnd. Con esta finalidad sale este nuevo servicio Azure Static Web Apps que hoy aún está en Preview, con lo cual es posible que alguna de las opciones que se comenta en el artículo es posible que se modifiquen cuando estén en GA (disponibilidad general).

Introducción

Azure Static Web Apps es un servicio que compila e implementa automáticamente aplicaciones web en Azure desde un repositorio de GitHub.

¿Qué es una Static Web Apps? Son aplicaciones creadas con un Frameworks JavaScript como pueda ser React, Angular o Vue por ejemplo y cuya compilación es un conjunto de HTML, CSS y JavaScript. Cuando esto lo hacemos desde un Web App o un servidor tradicional estos recursos estáticos se sirven desde un único punto y van unidos al resto de procesos de la aplicación. Cuando hacemos uso de este nuevo servicio estos recursos no están centralizados en un único sitio, sino que están distribuidos geográficamente en todo el mundo (de una forma similar a un Common Delivery Network). Esta distribución permite ofrecer archivos de forma más rápida ya que se encuentran más cerca de los usuarios.

"un servicio que compila e implementa automáticamente aplicaciones web en Azure desde un repositorio de GitHub"

Este servicio está pensando desde el punto de vista de un desarrollador y es en el momento en el que se realice algún cambio en la aplicación este automáticamente se despliegue. Por este motivo en el momento en el que se crea un commit sobre nuestro repositorio de GitHub se lanza un

proceso (GitHub Actions) que publica el código en Azure. Más adelante veremos que son las GitHub Actions y el proceso que llevamos.

Que tipos de aplicaciones podemos implementar con este servicio:

- Crear Apps modernas con JavaScript usando Frameworks como Angular, React, Vue y un backend con Azure Functions.
- Publicar sitios web estáticos con Frameworks como Gatsby, Hugo o VuePress.
- Implementar Aplicaciones web con Frameworks como Next.js y Nuxt.js.

El éxito de este servicio radica principalmente en que este tipo de aplicaciones son las que más se desarrollan, casi toda aplicación tiene un frontal web haciendo uso de JavaScript. Y en ese punto tenemos dos escenarios:

- Frontales Web.
- Plataformas de Blogs en creciente auge debido a que muchos creadores de contenido buscan una plataforma mucho más ligera.

¿Qué características tiene?

- Hospedaje web para contenido estático como HTML, CSS, JavaScript e imágenes.
- Compatibilidad con la API integrada proporcionada por Azure Functions.
- Integración propia de GitHub, en la que los cambios en el repositorio desencadenan compilaciones e implementaciones. Actualmente no tiene integración con Azure DevOps aunque se puede hacer un pipeline para integrarlo.
- Contenido estático distribuido globalmente, lo que permite que el contenido esté más cerca de los usuarios.
- Certificados SSL gratuitos, que se renuevan automáticamente.
- Dominios personalizados* para proporcionar personalizaciones de marca a la aplicación.
- Modelo de seguridad ágil con un proxy inverso al llamar a las API, lo que no requiere ninguna configuración de CORS.
- Integraciones del proveedor de autenticación con Azure Active Directory, Facebook, Google, GitHub y Twitter.

- Definición de roles de autorización personalizables y asignaciones.
- Reglas de enrutamiento de back-end que permiten tener control total sobre el contenido y las rutas que atiende.
- Versiones de almacenamiento provisional generadas que se basan en las solicitudes de incorporación de cambios y permiten obtener versiones preliminares del sitio antes de su publicación.

Creación de una Static Web App con Hugo

Todo esto suena muy bien, pero vamos a ver un caso práctico y real. En este caso vamos a como poder montarnos un Blog haciendo uso de Hugo.

¿Qué es Hugo?

Hugo es un generador estático de páginas web HTML y CSS escrito en Go. Está optimizado para la velocidad, la facilidad de uso y la configurabilidad. Hugo toma un directorio con contenido y plantillas y las convierte en un sitio web HTML completo.

Hugo se basa en archivos Markdown con materia prima para metadatos, y puedes ejecutar Hugo desde cualquier directorio. está diseñado para funcionar bien en cualquier tipo de sitio web.

¿Cómo lo instalamos?

Para instalar Hugo lo haremos mediante un gestor de paquete del sistema operativo en el que estemos.

- En MacOS, puedes instalarlo con Homebrew: `brew install hugo`
- En Windows, con Chocolatey: `choco install hugo -confirm`
- Y en Linux, con snap: `snap install hugo`

En mi caso al trabajar en un entorno Windows he utilizado Chocolatey. Una vez ya está instalado en nuestro equipo verificamos que la instalación ha ido correctamente para ello tendremos que poner el siguiente comando:

```
C:\Users\vdiaz (blog@0.0.1)
λ hugo version
Hugo Static Site Generator v0.71.1/extended windows/amd64 BuildDate: unknown
```

Con Hugo instalado el siguiente paso es crearnos un sitio estático para ello tendremos que poner el siguiente comando:

```
C:\hugoCompartimos
λ hugo new site compartimos
Congratulations! Your new Hugo site is created in C:\hugoCompartimos\compartimos.

Just a few more steps and you're ready to go:

1. Download a theme into the same-named folder.
   Choose a theme from https://themes.gohugo.io/ or
   create your own with the "hugo new theme <THEMENAME>" command.
2. Perhaps you want to add some content. You can add single files
   with "hugo new <SECTIONNAME>\<FILENAME>.<FORMAT>".
3. Start the built-in live server via "hugo server".

Visit https://gohugo.io/ for quickstart guide and full documentation.
```

Como bien indica cuando hemos creado el sitio el siguiente

paso es elegir el tema que va a tener nuestro sitio para ello ir a la página <https://themes.gohugo.io> y elegimos uno entre todas las opciones que hay, en mi caso voy a seleccionar el template Soho. Para instalar un template en primer lugar lo que deberemos es descargarlo desde el GitHub donde esta y tenerlo ubicado en la carpeta de Themes

```
C:\hugoCompartimos\compartimos
λ cd themes\

C:\hugoCompartimos\compartimos\themes
λ git clone https://github.com/alexandrevicenzi/soho.git
Cloning into 'soho'...
remote: Enumerating objects: 627, done.
remote: Total 627 (delta 0), reused 0 (delta 0), pack-reused 627R
Receiving objects: 100% (627/627), 1.33 MiB | 688.00 KiB/s, done.
Resolving deltas: 100% (290/290), done.
```

En todo sitio de Hugo tiene un fichero `config.toml` que es en el que se configura cual es el tema que utiliza y donde se configura algunos de los aspectos propios del sitio. En nuestro caso en la primera línea de este fichero deberemos de indicar que nuestro Tema es Soho y añadir la siguiente información propia del template. El nuestro quedaría de la siguiente forma:

```
1 theme = "soho"
2 baseURL = "https://example.com"
3 title = "Compartimos"
4 languageCode = "en"
5 enableInlineShortcodes = true
6
7 summaryLength = 10
8 enableEmoji = true
9
10 [params]
11 author = "Author Name"
12 description = "My Blog"
13
14
15 profilePicture = "images/profile.png"
16
17 copyright = "Author Name"
18 license = "CC BY-SA 4.0"
19 licenseURL = "https://creativecommons.org/licenses/by-sa/4.0/"
20
21 customCss = ["css/blog.css"]
22 customJs = ["js/blog.js"]
```

Para ver si el tema se ha aplicado correctamente pondremos en la línea de comando la siguiente instrucción:

```
C:\hugoCompartimos\compartimos
λ hugo serve -D
Building sites ... EN
-----
Pages           | 9
Paginator pages | 0
Non-page files  | 0
Static files    | 6
Processed images| 0
Aliases         | 1
Sitemaps        | 1
Cleaned         | 0
-----
Built in 63 ms
Watching for config changes in C:\hugoCompartimos\compartimos\{archetypes,content,data,layouts,static,themes}
Environment: "development"
Serving pages from memory
Running in Fast Render Mode. For full rebuilds on change: hugo server --disableFastRender
Web Server is available at http://localhost:1313/ (bind address 127.0.0.1)
Press Ctrl-C to stop

Change of config file detected, rebuilding site.
2020-08-15 10:48:46.500 +0200
Rebuilt in 46 ms

Change of config file detected, rebuilding site.
2020-08-15 10:49:35.499 +0200
Rebuilt in 54 ms
```

Cuando hagamos cualquier modificación en el sitio automáticamente se compilará y se visualizará de nuevo. Si en el navegador arrancamos la url `http://localhost:1313` veremos que se abre un sitio con el theme que hemos seleccionado, en caso de que no se vea correctamente deberemos de revisar el fichero de configuración que algo hemos configurado de forma errónea.

Una vez ya tenemos creado el sitio el siguiente paso es añadir cualquier contenido para ello ponemos el siguiente comando:

hugo new posts/hello-compartimoss.md

Luego en el fichero md y con el editor de texto que queramos escribimos el artículo o el contenido que queramos (esta libertad para elegir el editor que se requiera es una de las grandes ventajas que tiene Hugo) y arrancamos el servidor y se debería de visualizar algo como el siguiente:

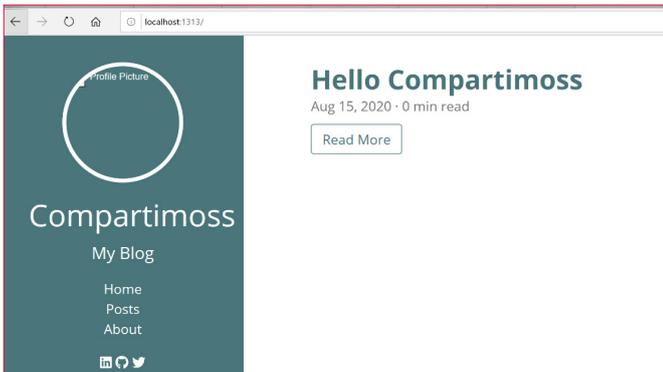


Imagen 1.- Visualización de la Web de Hugo.

Hugo está muy bien y nos da muchos beneficios, pero en este artículo hemos venido a hablar sobre el servicio de Azure Static Web Apps, para ello en primer lugar nos tendremos que crear un repositorio en GitHub y alojar el contenido donde está esta carpeta. En mi caso podéis ver el ejemplo en el siguiente repositorio. <https://github.com/AdrianDiaz81/compartimoss Hugo>

GitHub nos proporciona 5 repositorios privados donde podemos alojar nuestro blog sin que todo el mundo pueda visualizar nuestro contenido. En dicho ejemplo lo he dejado público para que los lectores puedan ver el flujo de aprobación el contenido de este.

Una vez tenemos el repositorio GitHub creado vamos a nuestra suscripción de Azure y creamos un servicio Azure Static Web Apps.

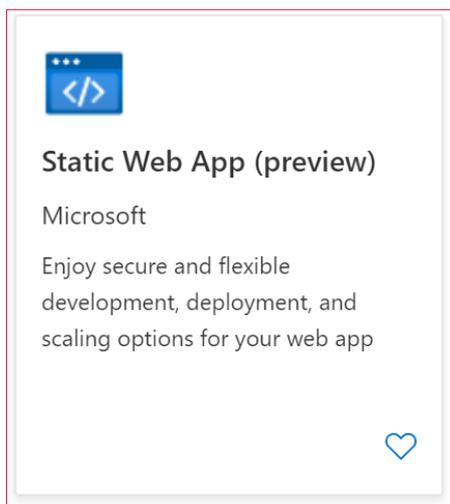


Imagen 2.- Servicio de Static Web App.

Cuando le damos a crear este servicio, al igual que en muchos servicios de Azure nos pedirá la región donde inicialmente estará este servicio, el grupo de recursos donde va a

estar alojado y nos pedirá que introduzcamos nuestras credenciales de GitHub, tal y como se muestra en la siguiente imagen:

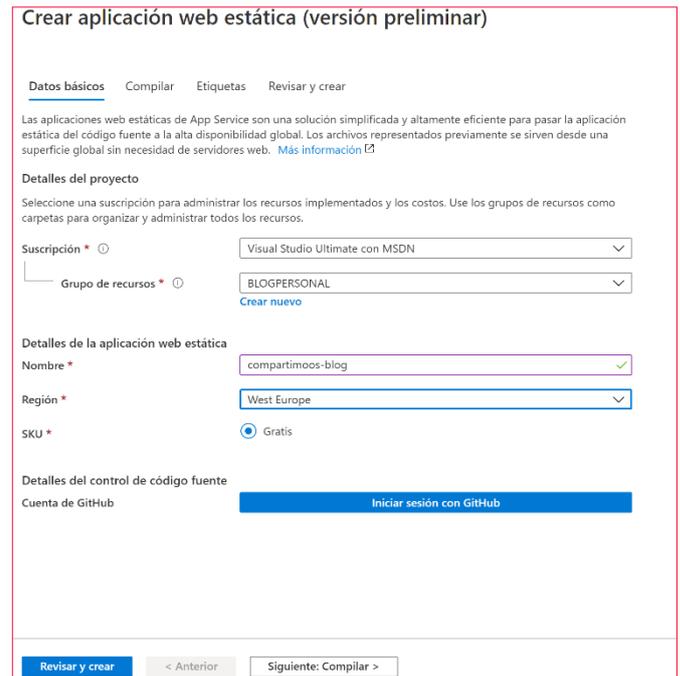


Imagen 3.- Configuraciones para la creación de la Static Web App.

Una vez iniciamos la sesión con GitHub y autorizamos a que el portal de Azure pueda leer el código de los repositorios. Tendremos que indicar cual es la Organización, el repositorio y la rama en la que vamos a obtener el código en nuestro caso el ejemplo sería el siguiente:



Imagen 4.- Configuraciones relativas a GitHub.

Con esta información le damos a crear al Servicio. Una vez el servicio se ha creado automáticamente se crea una GitHub Actions. Para los que no estén familiarizados con este servicio de GitHub, podemos decir que es como una Build de compilación de Azure DevOps. Una Actions lo que nos sirve es para automatizar los flujos de trabajo para poder compilar, empaquetar o implementar nuestro desarrollo. En este caso queremos que cada vez que se añada cualquier modificación en nuestra rama master automáticamente se lance este proceso y se actualice el código de nuestra aplicación.

"Tendremos que indicar cual es la Organización, el repositorio y la rama en la que vamos a obtener el código"

Nada más que se crea el servicio, automáticamente se lanza esta GitHub Actions. Para ello dentro de nuestro repositorio de GitHub en la pestaña Actions podemos ver que se ha lanzado una. Esta Actions es un fichero con extensión

.yml que se ha creado en nuestro repositorio. Este fichero tiene una única tarea que es compilar y ejecutar la aplicación. La primera que se lanza se produce un error debido a que falta por configurar la ubicación donde se encuentra el resultado de la compilación. Otro de los errores que se producen es donde está la ubicación de nuestro servidor Hugo. Si veis el fichero yml tiene unos parámetros en los que se indica dónde está la API si la hubiera (en este caso no hace falta), la ruta del código fuente y la ubicación de los artefactos. En mi caso los parámetros de configuración los he dejado de esta forma:

```

12 jobs:
13   build_and_deploy_job:
14     if: github.event_name == 'push' || (github.event_name == 'pull_request' && github.event.action != 'closed')
15     runs-on: ubuntu-latest
16     name: Build and Deploy Job
17     steps:
18       - uses: actions/checkout@v2
19       with:
20         submodules: true
21       - name: Build And Deploy
22         id: builddeploy
23         uses: Azure/static-web-apps-deploy@v0.0.1-preview
24       with:
25         azure_static_web_apps_api_token: ${{ secrets.AZURE_STATIC_WEB_APPS_API_TOKEN_BLACK_BAY_066FAD103 }}
26         repo_token: ${{ secrets.GITHUB_TOKEN }} # Used for Github integrations (i.e. PR comments)
27         action: "upload"
28         ##### Repository/Build Configurations - These values can be configured to match you app requirements. #####
29         # For more information regarding Static Web App workflow configurations, please visit: https://aka.ms/swaworkflowconfig
30         app_location: "/" # App source code path
31         api_location: "/api" # Api source code path - optional
32         app_build_output_directory: "public" # Built app content directory - optional
33         ##### End of Repository/Build Configurations #####

```

Conclusión

Azure Static Web Apps es de esos servicios que los desarrolladores estamos esperando que salga en alta disponibilidad, da un abanico de posibilidades muy fácil y sencillo, viene a cubrir una parte que ahora mismo estaba sobre

dimensionada dentro del catálogo de servicios PaaS que nos proporciona Azure. Sin embargo, para que este servicio tenga un mayor abanico de usuarios faltaría una integración con Azure DevOps, y no solo con GitHub. También queda pendiente ver los planes que nos ofrece y los precios de estos.

"una gran alternativa para alojar un blog y tener el contexto de todo lo que en él te hace falta"

Desde el punto de vista de posibilidades que nos da este servicio, me parece una gran alternativa para alojar un blog y tener el contexto de todo lo que en él te hace falta. En mi caso, hice la prueba para alojar mi blog personal y además de conocer Hugo como una plataforma nueva, este servicio te da la opción de integrarte en el mundo de las GitHub Actions y su integración con Azure.

ADRIÁN DIAZ CERVERA

Architect Software Lead at Encamina

MVP Office Development

<http://blogs.encamina.com/desarrollandosobresharepoint>

adiaz@encamina.com

@AdrianDiaz81





17

Entrevista Mayra Badillo Villamizar

Mi nombre es Mayra Badillo Villamizar, soy de Cúcuta y vivo en Bogotá (Colombia). En este último año he sumado a mi diario vivir el trabajar con subsidiarias de Microsoft en Latam y socios de negocio en España atendiendo los procesos de transformación digital en clientes que buscan modernizar sus procesos a través de Microsoft 365 y Dynamics 365 CRM.

Estudié Ingeniería de Sistemas en la Universidad Francisco de Paula Santander, he realizado estudios de BPO con la Universidad Javeriana en Colombia y Universidad de Belgrano Argentina. Inicé trabajando con SAB Miller soportando SAP, posterior a ello trabajé 5 años en Telefónica siendo recurso de Hewlett Packard para administrar la operación de TI en la sede de mi ciudad natal (Cúcuta).

Siendo empleada, inicié mis estudios en plataformas Microsoft (Windows Server, Exchange Server y Share Point Server), llevando mi sueño de ser freelance y emprendí como un Arquitecto de Soluciones atendiendo clientes en mi ciudad (tuve mi primera empresa con la cual trabajamos cableados estructurados e implementaciones de Di-



rectorios Activos y SharePoint Server para Windows Small Business Server, Asterisk y Cisco). En el 2010 viajé a Bogotá solo para conocer a Steve Ballmer y allí fue amor a primera vista con BPOS y Dynamics CRM Online.

En el 2011 me enfoqué como Freelancer trabajando con el BID (Banco Interamericano de Desarrollo) y realicé mi segundo emprendimiento, siendo Microsoft Partner ya desde el 2007. En ese año me convertí Microsoft Student Partner y me tracé como meta ser un Microsoft SWAT.

En el 2007 inicié con las comunidades Microsoft IT Pro

¿Por qué y cómo empezaste en el mundo de la tecnología?

Bien particular, de niña mi sueño y pasión era estudiar astrofísica (aún conservo mis telescopios y libros de Astronomía). Estando en el Colegio (hermosa historia de mi vida, colegio católico de señoritas) tuve la convicción de que sería Ingeniería y que quería un trabajo donde no sintiera el tedio de la monotonía, siempre he tenido conflictos de autoridad (detesto los uniformes y el modelo de tener que cumplir horario e ir a la misma oficina todos los días de la vida y tener una rutina). Así que entre los libros de mi tío Tim Osswäld estuve influenciada con la ingeniería además de los viajes y las experiencias de trabajo en alto rendimiento (mi tío es PhD en Polímeros, dirige un centro de investigación en Wisconsin y viaja entre Alemania, India

e Inglaterra además de otros países, se imaginarán la tremenda influencia en mi vida). Todos esperaban que estudiara periodismo o derecho, pues soy una persona obsesa de la lectura y de escribir (influencia de pequeña de una de las mujeres más especiales de mi vida además de mi mami, hermana y sobrina; mi querida y siempre recordada Oma, tía Ross Osswäld).

Así que estando en último año del colegio decidí seguir la seducción de las matemáticas y la tecnología y elegí Ingeniería de Sistemas (pero me matriculé pensando en ambientes industriales y comencé mi pasión con la Robótica y Automatización, lo que luego se trasladó a las Redes y Servidores). Existía un departamento en la universidad que administraba el centro de datos y redes y permitían tener becas-trabajo, escuché que era difícil entrar y era algo TOP, fue suficiente para mí el reto e ingresé; allí me

formé en Networking (sí, trabajé Linux y Solaris) hasta que me enamoré de Microsoft.

Y hoy, heme aquí, libre como lo soñé (sin el modelo tradicional de las empresas), viajando por diferentes países siendo speaker internacional y trabajando con clientes y subsidiarias. Con la libertad de elegir convertir un miércoles en sábado o con la pasión de estar días enteros trabajando por el infinito amor que mueve mi espíritu trabajando diseñando soluciones para clientes o siendo su consultora en la gestión del cambio y transformación digital. Amo los hoteles, amo viajar y disfrutar de un moscow mule en el bar del hotel, de pasar horas en la piscina, conocer nuevas culturas y de experiencias culinarias; además de visitar museos. Tengo el trabajo soñado, solo me falta 1, y es ser FTE Corp o LATAM de Microsoft. Tengo el resto de mi vida para seguir disfrutando esta vida que elegí.

¿Cuáles son tus principales actividades tecnológicas hoy en día?

Desde Junio soy Microsoft Most Valuable Professional (MVP) Business Applications. Desde enero 2020 Microsoft Subject Matter Expert como Field Enablement Coach trabajo con equipos internos en diferentes subsidiarias de Microsoft Latam a través del WW Learning.

Como SWAT SENIOR me desempeño en el rol de Arquitecta en Productividad, Colaboración, Seguridad y BizAPPs en Microsoft 365 y Power Platform apoyando el ecosistema de socios de negocios, mayoristas y clientes finales de acuerdo con la estrategia vigente a trabajar en un Fiscal Year. Speaker Internacional. Microsoft Partner Services desde 2008. Freelancer con cobertura en clientes (Microsoft, Partners, Mayoristas y clientes finales) de Centroamérica, Suramérica y España. Arrancando año fiscal Microsoft me encuentro atendiendo clientes de la vertical Hidrocarburos, Laboratorio Farmacéutico, Construcción, Retail y Banca; además de proyectos en el Caribe, Centro América y Cono Sur. Cerramos el FY20 con una consultoría para 2 Aerolíneas así como un cliente en España-Francia.

Microsoft SWAT TEAM (7 años desde el 2013) en productividad basada en Microsoft 365, Modern Workplace, Project Online, Business Applications y Power Platform.

Consultora en Gestión del Cambio OCM - Prosci (modelo Microsoft de servicio) para la adopción y usabilidad de soluciones basadas en Microsoft Modern workplace y Business Applications en Power Platform. En mis ejercicios profesionales religiosamente aplico prácticas basadas en Arquitectura Empresarial TOGAF, CONSITIO en MOF, ITIL, Surestep on Dynamics, BPM y BABOK.

¿Cuáles son tus principales actividades NO tecnológicas hoy en día?

Como actividades, mentiría; ya que no hay nada en mi vida ajeno al ámbito tecnológico. Pero como principales activi-

dades de vida hoy en día está el disfrutar de mi familia (papá, hermana y sobrinos (darles tiempo de calidad y atención), ser una dichosa tía ☺ además de disfrutar y compartir con mis mejores amigos a quienes considero hermanos. Espero poder escribir el libro que tengo planeado por ahora complemento mis días con el descubrimiento personal, desarrollo para el crecimiento profesional, tenía planeado iniciar éste 2020 con un Master o MBA así como por fin ir a mi primer de muchos viajes a mi amada y anhelada Alemania de mi corazón.

¿Cuáles son tus hobbies?

Mi apartamento está lleno de gadgets, disfruto cocinar; amo leer e investigar acerca de acontecimientos históricos; cultivo mucho el conocimiento y aprendizaje de una de mis épocas favoritas de la historia, de la cual parte de mis convicciones del mundo y la vida tienen su génesis. Algunos gastan su dinero en eventos deportivos, yo lo hago en conciertos a los que voy sola o con mis mejores amigos (en septiembre viajaba solo para ver a Rammstein), sigo esperando poder ver AC/DC en vivo y repetir las veces que pueda U2. Disfruto la fotografía amateur y soy melómana empedernida (prefiero que se me quede el celular, pero lo único que me hace regresar si se olvida, es mi iPod). Una de mis pasiones es viajar y disfrutar de hoteles y conocer nuevos restaurantes además de ir a un Pub, ya sea por cerveza, sidra o unos gintonics o moscow mule. Disfruto ir sola a cine, elegir una exquisita película en sala premier pedir sushi; lo que me lleva a otra pasión, podría comer sushi todos los días de mi vida, por lo que Japón es otro destino pendiente. Espero poder hacer en algún momento de mi vida (pues me considero una workaholic cosa que disfruto) hacer 1 año sabático al mejor estilo de comer (New York y Gran Bretaña), rezar (Japón) y amar (Alemania)

¿Cuál es tu visión de futuro en la tecnología de acá a los próximos años?

En los próximos años veremos cómo gracias a la democratización de la IA y el ML, que en el caso particular de Microsoft se cuenta gracias a la Power Platform, serán los information workers quienes desarrollen las soluciones que transformen más que el negocio, las áreas por las que responden, siendo influenciadores de la innovación no solo de sus puestos de trabajo sino de sus operaciones.

Mi visión, además, es que en el futuro a corto plazo más empresas abrirán sus modelos de trabajo hacia el home office olvidando por fin que a los colaboradores se les mide por el horario y el tiempo sentado en el mismo lugar de todos los días; y permitiendo que a los colaboradores se les evalúa y se les gestiona por sus resultados, sus entregables.

Y por último, en los próximos años veremos una dinámica acelerada donde las personas contarán más con sistemas

predictivos e inteligentes para la toma de decisión por no humanos; pero el desafío está en cuanto podrían disminuir los costos de TCO que esta tecnología demanda e implica; cerrando mi apreciación con que estamos ya ad portas de nuevas generaciones donde las brechas y curvas de adopción son más pequeñas por lo que hay gran oportunidad para los procesos de gestión del cambio, adopción y usabilidad de la tecnología.

MAYRA BADILLO VILLAMIZAR

Microsoft Subject Matter Expert | Microsoft MVP Business Applicatios

@mabvill

<https://www.linkedin.com/in/mabvill/>

próximamente www.mabvill.com

En **encamina** buscamos:

- ★ Desarrolladores .NET
- ★ Desarrolladores Dynamics 365
- ★ Consultores Office 365
- ★ Consultores CRM
- ★ Consultores de Azure

Si tú también **piensas en colores**



¡Queremos tu talento!
rrhh@encamina.com

encamina

[@encamina](#) [f](#) ENCAMINA [in](#) ENCAMINA



Gestión de servidores externos mediante Azure Policy

Si actualmente estás trabajando con Azure, probablemente hayas escuchado hablar de las Políticas de Azure. Digamos que son elementos clave para gobernar toda la infraestructura y servicios desplegados sobre Azure, que los podamos auditar y en cierta manera, forzarnos a cumplir el compliance que algunos estándares nos “obligan” a hacer.

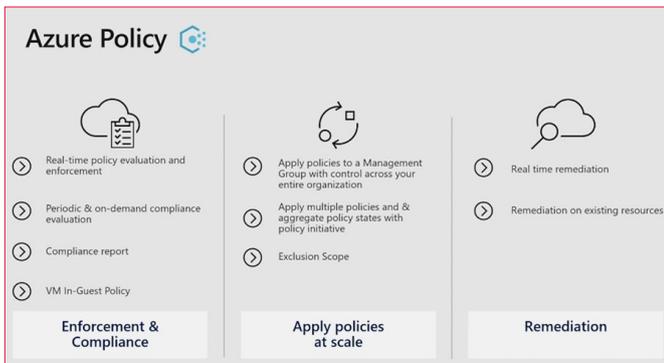


Imagen 1.- Detalles de Azure Policy.

Por ejemplo, el hecho de solo permitir la creación de VMs en ciertas regiones, o bien que solo se pueda utilizar un SKU determinado (muy útil para entornos de desarrollo donde no queremos disparar costes), restringir la creación de IPs Públicas, forzar la aplicación de valores, y un largo etc.

"Azure Policy son elementos clave para gobernar toda la infraestructura y servicios desplegados sobre Azure"

En este artículo no vamos a cubrir los básicos de Azure Policy, ya que hay infinidad de contenido por Internet que lo explicarán mejor que yo.

Dentro del portal de Azure, hay algunas políticas que nos permiten que automáticamente las VM's desplegadas en nuestra suscripción, se les implemente el agente de monitorización, de manera que tengamos el entorno automatizado y totalmente controlado (os suena a CI/CD, ¿no?), pues bien, lo que vamos a hacer en este artículo es que mediante el uso de las Políticas de Azure y Azure Arc vamos a mostrar la capacidad que tenemos de monitorizar no solamente las máquinas que tengamos en nuestras suscripciones de Azure, sino también las de otros proveedores Cloud.

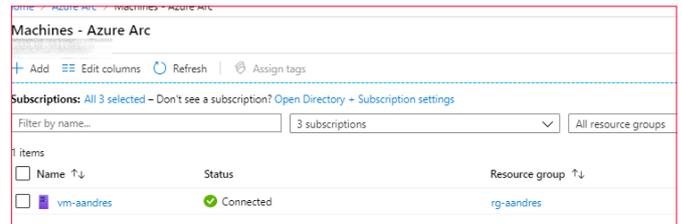


Imagen 2.- Gestión de Servidores con Azure Arc.

El uso de las Guest Policy, nos brindará la posibilidad de monitorizar los sistemas y las políticas de seguridad en todas las VM's. Teniendo de esta forma un plan de gobierno completo, tanto para sistemas Azure como externos (AWS, GCloud, Alibaba, etc...)

Configuración de la gestión de servidores “invitados”

Para poder empezar con ello, debemos de habilitar el proveedor de “Microsoft.GuestConfiguration” en la suscripción que pertoque. Este es el endpoint que permite la gestión de los servidores “invitados”:

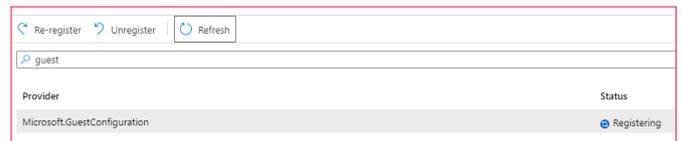


Imagen 3.- Habilitar el proveedor para gestionar servidores “invitados”.

Creación de la iniciativa

Siguiente paso es ir a las políticas de Azure y buscar las definiciones que tenemos disponibles, en este caso, buscamos una iniciativa (lo que viene siendo un conjunto de definiciones de políticas), en este caso buscamos la siguiente:

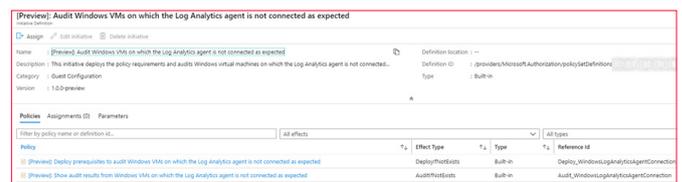


Imagen 4.- Definiciones de Políticas disponibles.

Vemos que hay dos definiciones dentro de la iniciativa que tienen efectos diferentes, por lo que vamos a darle un vistazo con más detalle a cada una de ellas. Lo primero que podemos observar, es que este tipo de implementaciones utilizan los workspaces de LogAnalytics, por lo que estaremos enviando telemetría de las VMs a los WS, que a posteriori nos servirán para pintar la información que queramos

de una forma muy intuitiva, permitiendo la lectura de la información a cualquier persona (aunque no sea techie).

LogAnalytics da una flexibilidad brutal dentro de Azure, este post no lo trataremos, pero veremos que lo mencionaremos en algún que otro momento, sobre todo para esa monitorización esencial en Azure (otro pilar fundamental del gobierno de Azure). La primera de ellas, podemos observar que su target de VM's son aquellas que tenemos o bien dentro de la suscripción de Azure, o bien aquellas que estén en otras plataformas Cloud y estén gestionadas por Azure Arc:

```

23 }
24 "policyRule": {
25   "if": {
26     "anyOf": [
27       {
28         "allOf": [
29           {
30             "field": "type",
31             "equals": "Microsoft.Compute/virtualMachines"
32           },
33           { ... }
34         ]
35       },
36       {
37         "allOf": [
38           {
39             "field": "type",
40             "equals": "Microsoft.HybridCompute/machines"
41           },
42           { ... }
43         ]
44       }
45     ]
46   }
47 }

```

Si vamos un poco más allá en la definición del JSON, podemos ver que:

- Se realiza una asignación para la configuración del agente Guest.
- Se crea una identidad administrada para la VM en cuestión.
- Se añade una extensión llamada "Guest Configuration".

Aquí no debemos de confundirnos, estas configuraciones solo se implementan a nivel de Azure, no a nivel de la VM como tal. Realmente, esto lo que implementa es una plantilla ARM a Azure

```

273 "condition": "[equals(tolower(parameters('type')), tolower('Microsoft.Compute/virtualMachines'))]",
274 "apiVersion": "2018-11-20",
275 "type": "Microsoft.Compute/virtualMachines/providers/guestConfigurationAssignments",
276 "name": "[concat(parameters('vmName'), '/Microsoft.GuestConfiguration/', parameters('configurationName'))]",
277 "location": "[parameters('location')]",
278 "properties": {
279   "guestConfiguration": {
280     "name": "[parameters('configurationName')]",
281     "version": "1.*",
282     "configurationParameter": {
283       "name": "[LogAnalyticsAgentLogAnalyticsAgentWorkspaceId]",
284       "value": "[parameters('workspaceId')]"
285     }
286   }
287 },
288 "condition": "[equals(tolower(parameters('type')), tolower('Microsoft.Compute/virtualMachines'))]",
289 "apiVersion": "2017-03-30",
290 "type": "Microsoft.Compute/virtualMachines",
291 "identity": {
292   "type": "SystemAssigned"
293 },
294 "name": "[parameters('vmName')]",
295 "location": "[parameters('location')]"
296 },
297 {
298   "condition": "[equals(tolower(parameters('type')), tolower('Microsoft.Compute/virtualMachines'))]",
299   "apiVersion": "2015-05-01-preview",
300   "name": "[concat(parameters('vmName'), '/AzurePolicyforWindows')]",
301   "type": "Microsoft.Compute/virtualMachines/extensions",
302   "location": "[parameters('location')]",
303   "properties": {
304     "publisher": "Microsoft.GuestConfiguration",
305     "type": "ConfigurationforWindows",
306     "typeHandlerVersion": "1.4"
307   }
308 }

```

Mientras que para aquellos servidores que están gestionados mediante Azure ARC, solo se implementa un recurso en Azure:

- Asignación de la configuración de "invitado".

```

Definition Assignments (0) Parameters
247     type : string
248     "workspaceId": {
249       "type": "string"
250     }
251   },
252   "resources": [
253     {
254       "condition": "[equals(tolower(parameters('type')), tolower('microsoft.hybridcompute/machines'))]",
255       "apiVersion": "2018-11-20",
256       "type": "Microsoft.HybridCompute/machines/providers/guestConfigurationAssignments",
257       "name": "[concat(parameters('vmName'), '/Microsoft.GuestConfiguration/', parameters('configurationName'))]",
258       "location": "[parameters('location')]",
259       "properties": {
260         "guestConfiguration": {
261           "name": "[parameters('configurationName')]",
262           "version": "1.*",
263           "configurationParameter": {
264             "name": "[LogAnalyticsAgentLogAnalyticsAgentWorkspaceId]",
265             "value": "[parameters('workspaceId')]"
266           }
267         }
268       }
269     }

```

Estas son las partes fundamentales de la policy que forma parte del 50% de la iniciativa, la otra policy que tenemos incluida en la iniciativa, es mucho más sencilla, ya que únicamente se dedica a pintar la información extraída por la otra policy.

Implementación de la policy

Una vez explicados los básicos, podemos implementar la policy, para ello, como he comentado al principio del artículo, vamos a necesitar el ID de un workspace de LogAnalytics, ya que ahí, vamos a ingestar toda la telemetría, y será nuestra fuente de datos, para su uso posterior.

"las policy nos pueden ayudar a mantener la cohesión de los recursos que tenemos en Azure"

En este caso, no voy a entrar en detalle en como configurar una iniciativa/Policy en Azure, pero lo que, sí que quiero remarcar, es un punto importante de esta iniciativa en concreto (y muchas de Azure), y es el apartado de remediación:

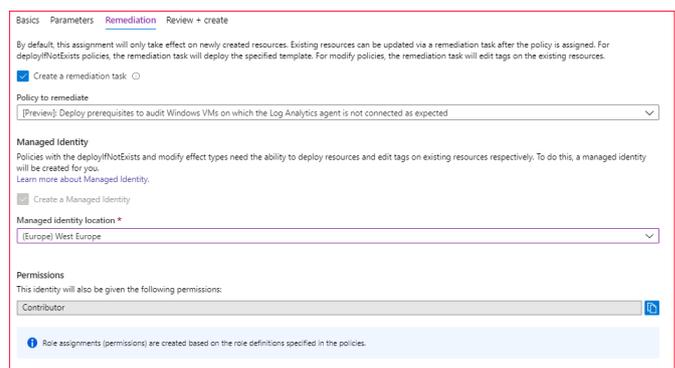


Imagen 5.- Sección de remediación.

Hay que tener en cuenta que estas policies, solo aplicarán a los recursos nuevos que se creen una vez que la Policy esté implementada y funcionando, pero no así para los recursos ya existentes, por lo que este apartado, cobra un especial interés.

Como anteriormente hemos visto en la definición, esta implementa todo por nosotros, por lo que recomiendo implementar una tarea de remediación, que nos permitirá implementar todos los recursos que sean necesarios para aquellos recursos existentes y que también nos interesa que entren dentro del alcance de la Policy. Recordad el parámetro "DeployIfNotExists" que teníamos en la definición

que irá ligado al Managed Identity, para aquellos recursos que sean necesario modificarlos.

"Gracias al uso de las tareas de remediación podremos relanzar nuestra policy y ver que se implementan los recursos necesarios para que cumpla la política"

Una vez implementada en el ámbito deseado, veremos el estado de esta en el portal de Azure y como irá cambiando el estado a medida que se vaya implementado y generando información sobre los recursos implementados en nuestra suscripción u otras plataformas Cloud.



Name	Effect Type	Compliance state	Non-Compliant Resources	Total resources
(Previous) Show audit results from Windows VMs on which the Log Analytics agent is not connected as expected	AuditIfNotExists	Non-compliant	12	17
(Previous) Deploy prerequisites to audit Windows VMs on which the Log Analytics agent is not connected as expected	DeployIfNotExists	Compliant	0	17

Imagen 6.- Visualización del estado de Políticas.

Gracias al uso de las tareas de remediación podremos relanzar nuestra policy y ver que se implementan los recursos necesarios para que cumpla la política y así podamos tener el entorno cohesionado y bajo los parámetros que nosotros necesitamos, tal y como podemos ver a continuación:



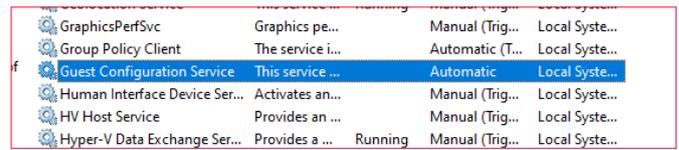
Name	Resource Group	Location	Created	Updated	Author
windowsloganalyticsagentconnection	resourcegroups/vm-aandres	4/6/2020, 2:11 PM	4/6/2020, 2:11 PM	Patricio12k/vp-aandres	
azurepolicyforwindows	resourcegroups/vp-aandres	4/6/2020, 2:11 PM	4/6/2020, 2:11 PM	Patricio12k/vp-aandres	
vm-aandres	resourcegroups/vp-aandres	4/6/2020, 2:11 PM	4/6/2020, 2:11 PM	Patricio12k/vp-aandres	

Imagen 7.- Recursos implementados.

Para cada tipo de definición que hemos visto anteriormente en el JSON, implementa una serie de recursos, por lo que nos permite visualizar que es lo que ha pasado y cuando ha pasado como tal.

¿Y qué hay de las máquinas incluidas en el ámbito?

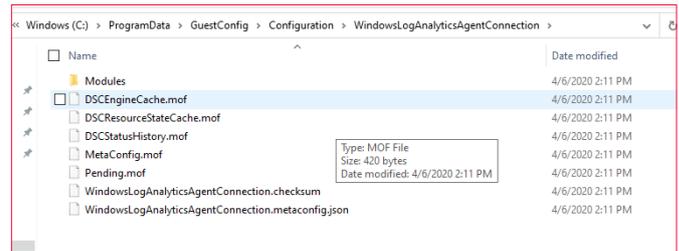
En este caso, el primer punto que debemos tener en cuenta es que se utiliza un servicio llamado "Guest Configuration Service", que es el responsable de las configuraciones:



Name	Description	Startup Type	Status
GraphicsPerfSvc	Graphics pe...	Manual (Trig...	Local Syste...
Group Policy Client	The service i...	Automatic (T...	Local Syste...
Guest Configuration Service	This service ...	Automatic	Local Syste...
Human Interface Device Ser...	Activates an...	Manual (Trig...	Local Syste...
HV Host Service	Provides an ...	Manual (Trig...	Local Syste...
Hyper-V Data Exchange Ser...	Provides a ...	Running	Local Syste...

Imagen 8.- Servicio Guest Configuration Service.

Por lo que una vez los servidores estén conectados, se descargarán las configuraciones pertinentes desde Azure. Estas las podemos encontrar en c:\ProgramData\GuestConfi\configuration



Name	Date modified
Modules	4/6/2020 2:11 PM
DSCEngineCache.mof	4/6/2020 2:11 PM
DSCResourceStateCache.mof	4/6/2020 2:11 PM
DSCStatusHistory.mof	4/6/2020 2:11 PM
MetaConfig.mof	4/6/2020 2:11 PM
Pending.mof	4/6/2020 2:11 PM
WindowsLogAnalyticsAgentConnection.checksum	4/6/2020 2:11 PM
WindowsLogAnalyticsAgentConnection.metaconfig.json	4/6/2020 2:11 PM

Imagen 9.- Configuraciones pertinentes.

Así mismo, podemos ver definiciones DSC, que será evaluadas continuamente, y que a posteriori, serán mostradas en el portal de Azure. En estos casos el reportar información contra el WS de Log Analytics, tarda cierto tiempo, hay que tener en cuenta que analiza y luego reporta todo, por lo que esto pueden ser minuto o incluso horas.

Con todo este procedimiento, hemos aprendido que las policy no solamente nos pueden ayudar a mantener la cohesión de los recursos que tenemos en Azure, sino que, con la ayuda de Azure Arc, podemos extender el gobierno que hayamos establecido en tu estrategia en el Cloud.

ALBERTO ANDRÉS
 Cloud Solutions Architect
 @albandrod

Un repaso a las herramientas de Microsoft para migrar a SharePoint Online y OneDrive For Business

Como parte de su estrategia de facilitar la transición a Microsoft 365 y sus servicios principales entre los que se encuentran SharePoint Online (SPO) y OneDrive for Business (ODFB), Microsoft proporciona un juego de herramientas de migración diseñadas para simplificar al máximo la carga de información desde orígenes On-Premises y Cloud a sitios de SPO y ODFB. En este artículo haremos un repaso al juego de herramientas de migración provistas por Microsoft sin coste.

Juego de Herramientas de Migración

Migración dispone de un juego de herramientas de migración que se compone de:

- La SharePoint Migration Tool o SPMT, primera herramienta de migración de Microsoft disponible desde 2017 y que permite migrar contenidos desde servidores de archivos o bien desde Granjas de SharePoint (versiones 2010, 2013 y 2016) a sitios de SPO y ODFB
- Migration Manager, un servicio de migración integrado en el SharePoint Admin Center que facilita migrar contenidos desde servidores de archivos a sitios de SPO y ODFB a partir de crear tareas de migración que son gobernadas y orquestadas por Migration Manager.
- Mover, un servicio cloud de migración adquirido por Microsoft en 2019 y que facilita la migración desde servicios de almacenamiento cloud (Como DropBox o Box) y/o On-Premises a Sitios de SPO y ODFB.

"La SharePoint Migration Tool o SPMT que permite migrar contenidos desde servidores de archivos o bien desde Granjas de SharePoint"

SharePoint Migration Tool

La SPMT fue la primera herramienta de migración desarrollada por Microsoft y que ha ido evolucionando en los últimos años incorporando nuevas funcionalidades. La PMT permite migrar contenidos desde servidores de archivos On-Premises o bien desde Granjas de SharePoint 2010/2013/2020 a Sitios de SPO y ODFB:

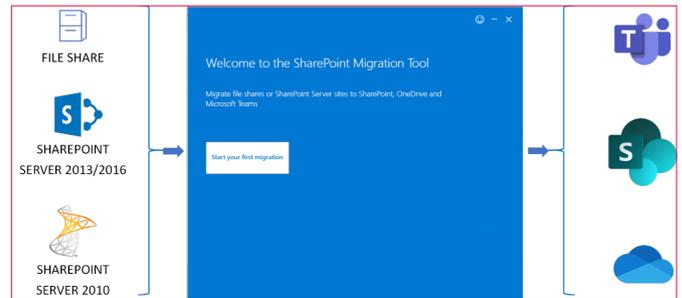


Imagen 1.- Orígenes y destinos de migración con la SPMT.

El resumen de las capacidades de la SPMT es el siguiente:

- Soporte para migraciones de tipo masivo a través de definir archivos de migración en formato JSON o CSV en el que se establecen los orígenes de migración y por cada origen los destinos correspondientes en SPO y ODFB.

#	A	B	C	D	E
1	C:\MigrationTest\testfiles			https://contoso.sharepoint.com/sites/Sample/	DocLibraryName OptionalSubFolder
2	https://sharepoint2013.com/sites/contoso/team1/	SourceDocLibrary	OptionalSubFolder	https://contoso.sharepoint.com/sites/Test/	Documents OptionalSubFolder
3	\\server\folder\sharepoint\test\images			https://contoso-my.sharepoint.com/personal/morgan@contoso.com/	Documents OptionalSubFolder

Imagen 2.- Ejemplo de archivo de migración en formato CSV.

- Granularidad en la configuración de la migración que nos permite realizar parametrizaciones como omitir ciertos contenidos en base a una condición de tipo fecha, establecer un mapeo de usuarios entre origen de migración y destino, etc.

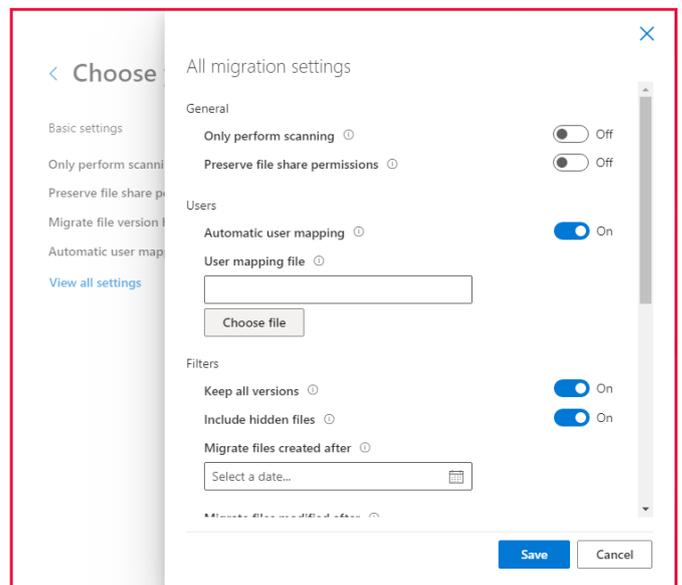


Imagen 3.- Opciones de configuración de la SPMT.

- Informes detallados en formato CSV de cómo ha transcurrido una migración concreta

- Realizar un assesment previo de migración en el caso en el que vayamos a migrar desde SharePoint 2010/2013 a SPO a través de utilizar el SharePoint Migration Assesment Tool (SMAT). SMAT es una herramienta de línea de comandos que nos permite analizar una Granja de SharePoint 2010/2013 para identificar qué elementos no se podrán migrar.
- Soporte de PowerShell a través de 8 comandos provistos para realizar la migración por línea de comandos en lugar de con la UI de la SPMT.



Imagen 4. Comandos PowerShell para realizar migraciones con la SPMT.

Aunque no se ha mencionado, la SPMT requiere ser instalada en un equipo que por un lado tenga “visibilidad” del origen de información a migrar y por otro disponga de la conexión a Internet necesaria para poder migrar a SPO y ODFB. Los requerimientos mínimos y recomendados (en verde) para el equipo en el que se instale la SPMT son:

SPMT Requirements

CPU: 64-bit 1.4 GHz 2 Core vs. **Quad Core**

RAM: 8GB vs. **16 GB**

Local Storage: 150 GB vs. **150 GB SSD**

Network Card: High-speed Internet Connection vs. **1 Gbps**

Operating System: Windows 7, Windows Server 2008 R2 or better vs. **Windows 10, Windows Server 2012 R2 or later**

.NET Framework: 4.6.2 or later

Imagen 5.- Requerimientos del equipo en el que se instale la SPMT.

Una pregunta que seguro que os estaréis haciendo es la relativa a qué puede migrar la SPMT y sobre todo qué velocidad de migración puedo esperar. Responder a la primera pregunta es sencillo ya que Microsoft tiene muy bien documentado que soporta y que no soporta al migrar con la SPMT a SPO y ODFB (+ Información: <https://docs.microsoft.com/en-us/sharepointmigration/what-is-supported-spmt>):

- A nivel de migración de contenidos en servidor de archivos, se soporta tanto la migración de archivos en rutas locales del equipo donde se ha instalado la SPMT como rutas de red compartidas.
- A nivel de migración desde SharePoint 2010/2013/2016, se soporta la migración:
 - Sitios completos.

- Navegación, Icono y Descripción del Sitio.
- Listas y Bibliotecas de Documentos.
- Archivos, Carpetas, Elementos de Lista y Versiones
- Permisos.
- Metadatos Administrados y Taxonomía.
- Características y Web Parts
- Páginas.

En cuanto a qué no se soporta, como os podéis imaginar se trata sobre todo de elementos de SharePoint On-Premises que la SPMT no puede migrar. En concreto:

- No se soportan todos los tipos de listas: + Información.
- No se soportan todos los Web Parts: + Información.
- No todas las características de Sitio están soportadas: + Información.
- Sólo migra sitios out of the box, es decir sin código personalizado o con herramientas de terceros.
- Sólo migra la navegación por defecto de Sitios.

En lo que a velocidad de migración que podemos esperar, la respuesta es que va a depender del escenario de migración que nos encontremos:

Type of metadata	Examples	Maximum
Light	ISO files, video files	2 TB/day
Medium	List items, Office files (~1.5 MB)	1 TB/day
Heavy	List items with custom columns, small files (~50 kb)	250 GB /day

Imagen 6.- Rendimiento esperado en función del escenario de migración.

Aunque Microsoft ha documentado el rendimiento esperado en función del escenario de migración, mi experiencia en migraciones de datos On-Premises a SPO usando herramientas de migración en general (Al final todas usan las mismas APIs) es la siguiente:

- La tasa de migración por hora puede ser variable dependiendo del momento en el que estemos migrando. Por ejemplo, si estamos migrando por la mañana en horario europeo tendremos una tasa de migración más baja comparado con la que se puede conseguir por las tardes o durante los fines de semana. Esto se debe al mecanismo de bloqueo (Throttling) que Microsoft tiene implementado en SPO y ODFB.
- La velocidad de migración típica al migrar desde servidor de ficheros a SPO y ODFB suele oscilar entre 20 y 30 GB/H por instancia de migración.
- La velocidad de migración se reduce mucho en el caso en el que el número de archivos a migrar es muy grande y el tamaño de los archivos es pequeño.

Si habéis llegado hasta este punto, seguro que alguno se preguntará: ¿Cómo incremento el volumen de información migrado por hora? La respuesta es que diseñando una arquitectura de migración que nos permita paralelizar la migración a través de disponer de varias instancias de migración. De esta forma, podremos incrementar la tasa de migración tantas veces como instancias tengamos.

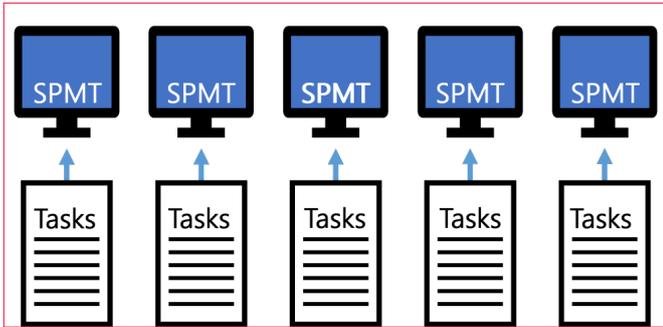


Imagen 7.- Arquitectura de migración con varias instancias de la SPMT.

Ahora bien, una arquitectura de migración como la representada en la Imagen 7 tiene varias problemáticas asociadas:

- Recursos necesarios (Hardware y Software) para el despliegue de la arquitectura de migración.
- Tiempo necesario para el setup de la arquitectura de migración.
- Ineficiencias debido a la carencia de capacidades de orquestación: cada instancia de migración se comporta de forma independiente en base a como se haya parametrizado la migración.
- Más susceptible a errores.

Precisamente, para dar respuesta a estos inconvenientes es donde el servicio de Migration Manager emerge como alternativa para realizar migraciones a escala.

Microsoft Migration Manager

Migration Manager es un servicio diseñado para realizar migraciones en las que se requiere capacidades de orquestación de tareas de migración, así como autoescalado y balanceo de estas. Migration Manager está pensado para escenarios de migración complejos en los que se requiere disponer de capacidades como:

- Gestión centralizada de tareas de migración.
- Balanceo de carga automático.
- Soporte para migraciones de tipo masivo.
- Reporting y monitorización.

Migration Manager habilita escenarios de migración que se fundamentan en la siguiente aproximación alto nivel:



Imagen 8.- Proceso de migración alto nivel con Migration Manager.

Como vemos, el proceso de migración consiste en:

- Puesta en marcha de la arquitectura de migración diseñada para la migración a realizar. Esta puesta en marcha consiste en descargar e instalar el agente de migración en cada una de las máquinas desde la que se realizará la migración (**Nota:** Los requerimientos Hardware y Software para el agente de migración son

los mismos que ya vimos para la SPMT). El proceso de instalación del agente es realmente sencillo:

- Descargar el paquete de instalación del agente.
- Seguir el asistente de instalación del agente en el que especificaremos por un lado la cuenta de acceso a SharePoint Online y por otro la cuenta de acceso al origen de migración.

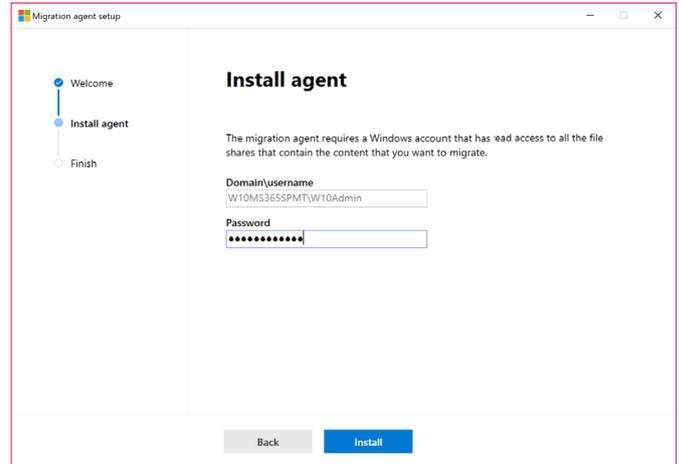


Imagen 9.- Asistente de instalación del agente de migración de Migration Manager.

- Verificar que el agente se ha instalado de forma correcta y aparece como un servicio de Windows que se inicia de forma automática.

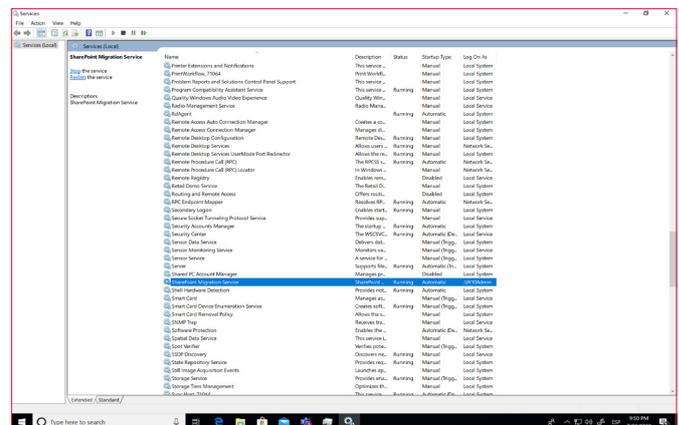


Imagen 10.- Servicio de migración instalado u operativo.

- Definir las tareas de migración necesarias para migrar información a SharePoint Online. Estas tareas de migración las definiremos en la Administración de SharePoint Online a través del correspondiente asistente que nos va guiando en su creación:

- Definir el método de migración de los archivos (**Nota:** A la fecha de redacción de este artículo, Migration Manager solo permite migrar archivos accesibles en una ruta compartida), donde podemos elegir entre una migración simple (un único origen y destino) o masiva (varios orígenes y destinos definidos en un archivo en formato CSV o JSON).
- En el caso de una migración simple, establecer el origen de migración de los datos que será una carpeta compartida a la que tiene acceso el servicio de migración.

- En el caso de una migración simple, establecer el destino de migración de los datos que será una ubicación de SharePoint en la que podremos elegir una biblioteca y un nivel de carpetas en la misma en el que realizar la copia de los archivos.
- Al igual que sucede con la SPMT, los ajustes finos de configuración de la migración que podamos requerir. Estos ajustes por defecto son comunes para toda tarea que se cree, pero luego a nivel particular de una tarea se pueden cambiar.

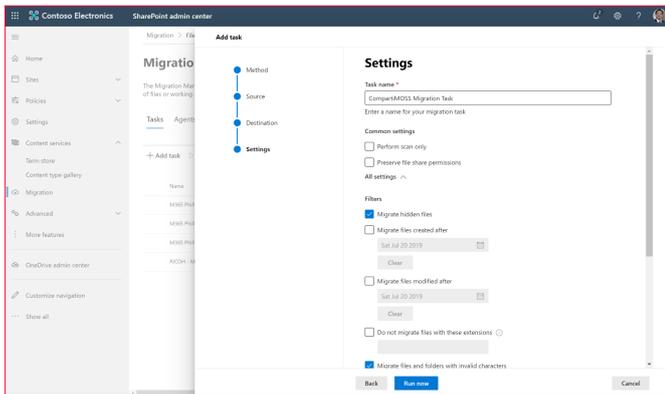


Imagen 11.- Ajustes en la tarea de migración.

A partir de que la tarea esté creada, es cuando la migración queda en manos de Migration Manager que se encarga de orquestar y autobalancear las tareas de migración en un escenario en el que tengamos varios agentes de migración instalados.

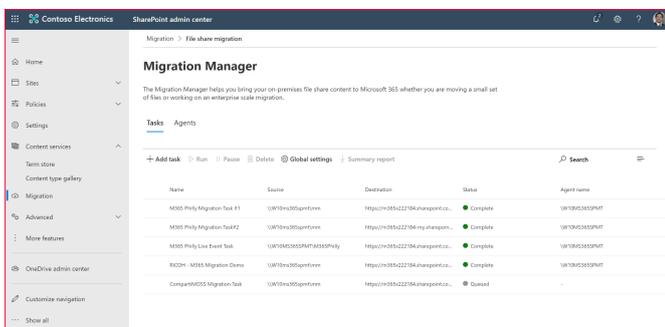


Imagen 12.- Tareas de migración en el Migration Manager.

- Monitorización y revisión: Por cada tarea tenemos la posibilidad siempre de ver el detalle de la misma lo que nos permite ir viendo su progreso e ir accediendo a informes parciales de la migración de manera que podemos ir controlando el proceso.

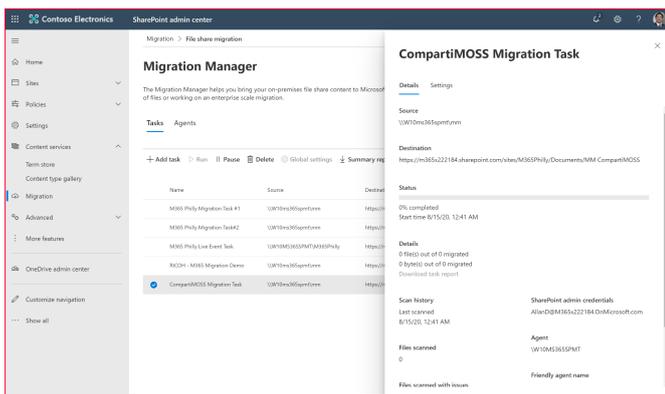


Imagen 13.- Detalle de una tarea de migración en Migration Manager.

Además, dada una Tarea, podemos seleccionarla en el listado de tareas de Migration Manager y realizar las siguientes acciones:

- Run: Volver a ejecutar la tarea si esta se ha completado de forma que lanzaremos la migración de nuevo y proporcionando soporte de migraciones de tipo incremental.
- Pause: Pausar la tarea seleccionada en el caso en el que se esté ejecutando.
- Delete: Borrar la tarea seleccionada.
- Summary report: acceder a un informe resumen de las acciones de migración realizadas por la tarea. Este informe es un archivo .csv.

Task name	Status	Source	Destination	Total scanned	Total to be Migrated	Items not migrated	Total bytes	Total GB	Migrated bytes	Migrated GB	Agent	Duration
CompartiMOSS Migration Task	Complete	\\W10ms3\https://m3	https://ms6522134.sharepoint.com/sites/MS65Phy/Document/M/MS65Phy	41	41	0	2.04E+09	1.9	2.04E+09	1.9	W10MS365	0:05:06

Imagen 14.- Informe de resumen de migración.

"Migration Manager un servicio de migración integrado en el SharePoint Admin Center que facilita migrar contenidos desde servidores de archivos a sitios de SPO y ODFB "

Mover

Mover es un servicio de migración ofrecido como una aplicación 100 nube que Microsoft adquirió en octubre 2019 y que puede ser utilizado por cualquier cliente de Microsoft 365 sin coste alguno. Mover está pensado para migrar contenidos entre servicios cloud (Aunque también permite migrar contenidos On-Premises) gracias al juego de conectores de migración de que dispone:

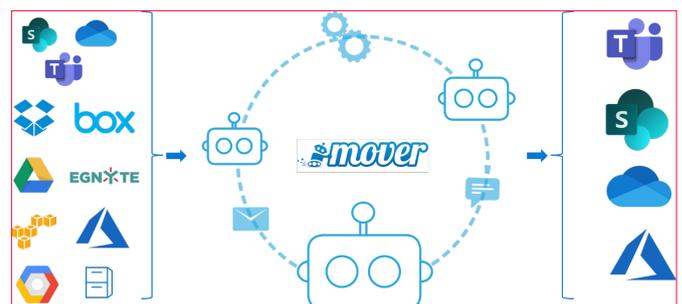


Imagen 15.- Migración de contenidos Cloud (y OnPremises) con Mover.

El resumen de las principales características de Mover (Para más información, os recomiendo revisar la documentación oficial de Mover: <https://docs.microsoft.com/en-us/sharepointmigration/mover-plan-migration>):

- No se requiere instalación alguna (Nota: Salvo en el caso en el que migremos contenidos de un servidor de ficheros): Todas las tareas relativas a configurar y lanzar una migración se realizan desde el navegador).
- Al igual que en el caso de la SPMT y de Migration Manager, podemos elegir entre realizar una migración

simple o bien realizar una migración de tipo masivo.

- Planificación y orquestación de migraciones a través de definir cuando se tienen que lanzar las tareas de migración de forma similar a como funciona Migration Manager.
- Logging en tiempo real de la migración, lo que nos permite monitorizar como se va realizando la migración.

Configurar Mover para realizar una migración Cloud To Cloud no es una tarea excesivamente compleja, aunque depende del origen y destino de migración. Por ejemplo, para realizar una migración de contenidos desde un Sitio de SharePoint en un tenant de Microsoft 365 a otro Sitio de SharePoint en otro tenant (Tenant to Tenant migration):

- Iniciamos sesión en Mover (<https://mover.io>) con nuestra cuenta de Microsoft 365.
- Definiremos en primer lugar el conector al origen de información. En este caso concreto, tendremos que elegir el conector de Office 365 y además darle los permisos necesarios en Azure AD. Este punto de los permisos os llevará seguramente a cometer errores, ya que tendréis que acceder a Azure AD para darle consentimiento de administrador a la App Office 365 Mover.

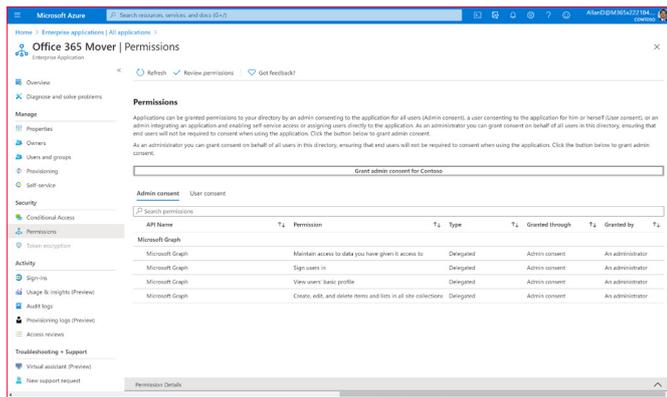


Imagen 16. Dando consentimiento de administrador a la App Office 365 Mover.

- De la misma forma, definiremos el conector al destino de migración. De nuevo, elegiremos el conector de Office 365 y realizaremos el consentimiento de los permisos correspondientes.
- Una vez que las conexiones están realizadas, podremos acceder al entorno de migración de Mover para crear tareas de migración:

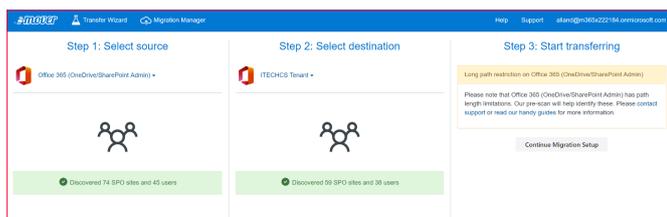


Imagen 17.- Conectores de Mover configurados.

- En el entorno de migración podremos:
 - Visualizar tareas de migración previas en el caso en el que se haya configurado para el origen y destino definidos. Para cada tarea definida podremos ver los detalles de esta y acceder a los informes disponibles.

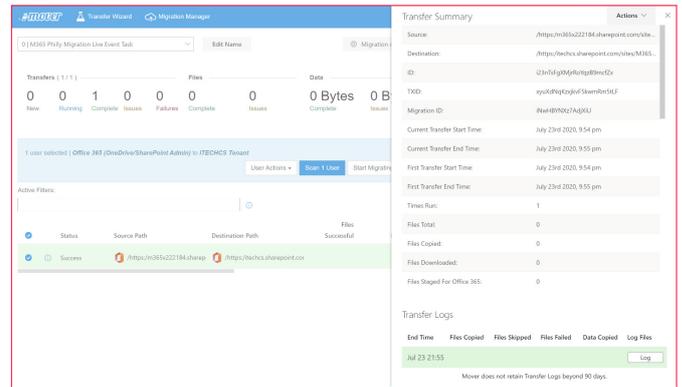


Imagen 18.- Ejemplo de detalle de tarea de migración realizada con Mover.

- Crear nuevas tareas de migración. Para cada tarea de migración tendremos la posibilidad de definirla de forma manual o bien realizar un escaneo automático en el que Mover establece los mapeos entre estructuras de origen y de destino. Este escaneo automático tiene sentido cuando migramos por ejemplo los OneDrive For Business entre dos tenants y contamos con la correspondencia necesaria entre usuarios en origen y en destino.

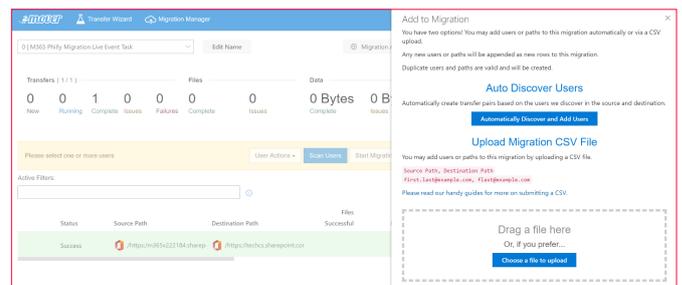


Imagen 19.- Creación de una nueva tarea de migración.

"Mover un servicio cloud de migración que facilita la migración desde servicios de almacenamiento cloud y/o OnPremises"

Además, dada una tarea de migración tendremos la posibilidad de tunearla antes de comenzar la migración. Por ejemplo, podremos hacer un ajuste fino de los contenidos de origen que vamos a migrar y también un ajuste fino del destino concreto donde vamos a mover la información como por ejemplo una carpeta concreta en la ruta de destino.

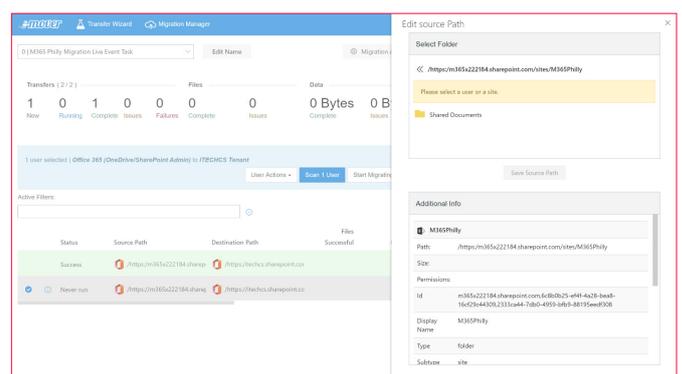


Imagen 20.- Ajuste fino del origen de contenidos a migrar.

- Lanzar la migración en el momento o bien planificar la misma:

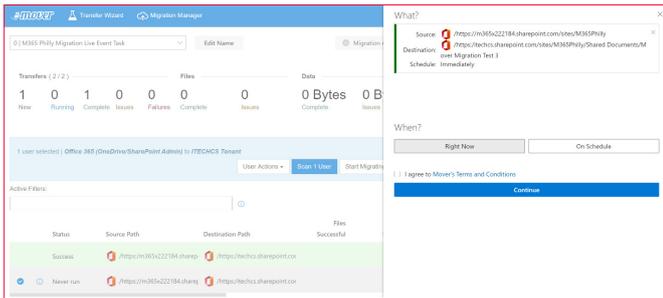


Imagen 21.- Lanzamiento de la migración en el momento vs. planificada.

- Acceder a reporting relativo a como se ha realizado la migración tanto en tiempo real como posteriormente descargando los correspondientes archivos CSV.

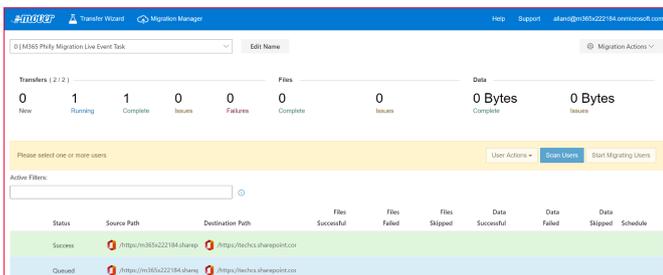


Imagen 22.- Estadísticas en tiempo real de una migración con Mover.

Conclusiones

A lo largo del artículo hemos revisado el juego de herramientas de migración de Microsoft que habilitan escenarios simples o complejos de migración sin coste adicional de licenciamiento, lo que permite acelerar la migración de contenidos On-Premises y Cloud a SharePoint Online y OneDrive For Business en Microsoft 365.

JUAN CARLOS GONZÁLEZ

Office Apps & Services MVP

M365 SME

@jcgm1978

Mentoring

Comparti MOSS

Un servicio experto alrededor de su SharePoint



CompartiMOSS le puede ayudar a través de su programa de Mentoring!

Contacte con nosotros y le enviaremos los planes de mentoring que tenemos disponibles para SharePoint.



Jupyter, Azure ML y Databricks

Cuando ayudamos a nuestros clientes a desarrollar proyectos de inteligencia artificial abrimos la puerta de la cocina, miramos debajo de la alfombra, detrás de la cocinilla, revisamos el filtro del extractor, etc. En definitiva, vamos a ayudar a esta empresa a implementar de principio a fin un ciclo de vida adecuado de desarrollo de software. Para esto, entre muchas actividades, participamos del proceso de desarrollo de los algoritmos y nos preocupamos por encontrar el mejor modelo para desarrollar entrenamientos de algoritmos y/o algoritmos que permitan hacer la inferencia de los modelos para realizar las predicciones adecuadas.

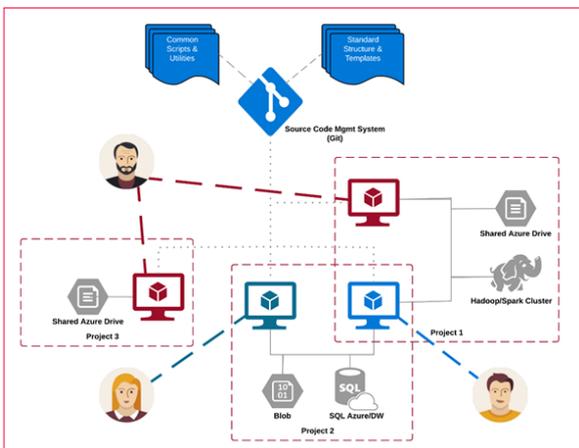


Imagen 1. Colaboración y desarrollo de algoritmos.

Lo habitual es encontrarnos con científicos de datos trabajando con notebooks en JupyterLab en local o en JupyterHub desde una máquina virtual. ¿Qué pensáis sobre esto? No tengo nada sobre JupyterLab y los notebooks, pero este modelo ofrece ciertos problemas y desventajas con riesgos que mitigar en una empresa:

- La seguridad de esa máquina virtual es importante, tengamos en cuenta que la conexión SSH suele estar abierta a internet y/o la autenticación se realiza con usuarios locales.
- Teniendo una máquina virtual, no nos aprovechamos de las ventajas del Cloud y nos toca mantener al día el sistema operativo, la versión de JupyterLab, copias de seguridad, etc.
- Este modelo de desarrollo puede ser aceptable para el trabajo de un científico de datos cuando está explorando la información con la que trabaja o probando ciertos algoritmos, visualizando los resultados y demás. ¿Qué pasa cuando queremos productivizar estos

algoritmos? ¿Programamos el entrenamiento en esta máquina virtual? ¿No necesitaremos un cluster con múltiples nodos para acelerar el entrenamiento y tener los resultados en un tiempo prudencial?

¿Qué alternativas tenemos? ¿Cómo podemos añadir seguridad a la vez que mejoramos el proceso de colaboración y desarrollo? Seguro que hay muchas opciones, pero, desde mi punto de vista, las más relevantes son Databricks y Azure Machine Learning, que os describo a continuación.

Databricks



Imagen 2. Características de Databricks.

Ante estas desventajas/problemas se creó Databricks, un servicio cloud que nos ayude a mejorar nuestro proceso de colaboración y desarrollo dando respuesta a estos de la siguiente forma:

- Databricks Workspace nos permite seguir trabajando con notebooks, pero con la seguridad empresarial necesaria, por ejemplo, con nuestras cuentas de Azure AD cuando estamos con Azure Databricks.
- Al ser un servicio PaaS, tenemos todas las ventajas que no nos ofrece una máquina virtual y es, en este caso el equipo de Azure, quien se encarga de las actualizaciones del producto, con nuevas funcionalidades o parches de seguridad.
- En Databricks Runtime tenemos un “self-service” de creación de clusters de Spark, lo que permite a cualquier usuario poder crear, arrancar, parar y ejecutar sus notebooks en un cluster, tan simple como seleccionar el cluster donde queremos hacer la ejecución desde el Notebook.
- El sistema de programación de tareas de Databricks, nos ayuda a productivizar los modelos, pudiendo realizar tareas programadas que ejecutan Notebooks en un cluster definido y dimensionado para la tarea que va a ejecutar. Esto no quita de tener que aplicar un buen modelo de ALM para que se ejecute los notebooks que tiene realmente el código de entrenamiento, evitando ejecutar, por ejemplo, las actividades de exploración y pruebas.

Azure Machine Learning

Aunque parece que Databricks es la opción ideal, la ventaja de la tecnología es que siempre hay diferentes opciones para realizar nuestras tareas que ofrecen diferentes puntos de vista frente a las otras opciones.

Microsoft ha ido evolucionando diferentes versiones de Azure Machine Learning y, en la última versión pública, podemos ver ciertas similitudes con las capacidades que desarrolló el equipo de Databricks buscando ayudar a mejorar el trabajo en los procesos de Big Data.

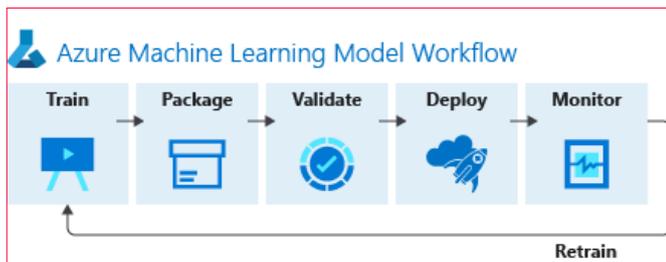


Imagen 3. Flujo de desarrollo de Azure Machine Learning.

Azure ML es un conjunto de servicios, herramientas, SDKs que permiten unificar el proceso de desarrollo de cualquier tipo de Machine Learning desde cualquier lenguaje de desarrollo o, incluso, sin código. Desde Azure ML se puede construir, entrenar y hacer un seguimiento de los modelos de Machine Learning.

Entre otros muchos servicios, Azure ML nos ofrece las siguientes capacidades que, desde mi punto de vista, permiten trabajar sin necesidad de tener una máquina virtual aislada con JupyterLab:

- Un Workspace donde podemos crearnos nuestra propia estructura de carpetas para almacenar nuestros Notebooks, además de la capacidad de poder editarlos utilizando un editor web o conectarlo a una instancia de computación donde poder ejecutar JupyterLab
- Un servicio de instancias de computación con nuestro propio "self-service" de creación de máquinas virtuales o clusters de entrenamiento. Estas máquinas virtuales utilizan de serie la autenticación de Azure AD para poder conectarnos a JupyterLab o RStudio que vienen de serie configurados y listos para desarrollar y conectarse desde los Notebooks.
- Una capa de abstracción a los orígenes de datos que nos ayuda a simplificar el código que tenemos que desarrollar para acceder a los orígenes típicos dentro de un proyecto de Big Data en Azure, Azure Storage, Azure Data Lake 2, etc.
- Un servicio de Pipeline para diseñar y programar las ejecuciones de nuestros entrenamientos, donde un ingeniero de datos puede implementar los flujos de entrenamiento ejecutando módulos con el código necesario.

¿Databricks o Azure ML?

Trabajar con Databricks nos va a implicar algún que otro

cambio en nuestro código para poder aprovechar la ejecución en el cluster de Spark que tenemos disponible para ejecutar nuestros notebooks. Nuestro código debería de estar optimizado para ejecutarse en Spark, y digo debería porque es la recomendación, ya que si ejecutamos código que no se puede ejecutar de forma clusterizada, esto es, que no se puede distribuir las tareas en los diferentes nodos de computación que podemos tener disponibles, las tareas de este notebook se ejecutarán en el nodo de Driver y no es los nodos Worker, con lo que trabajaremos únicamente con una instancia de computación del cluster que no está optimizada para la ejecución de las tareas.

Es posible que para el día a día de un científico de datos esto de Spark no sea lo óptimo, que las pequeñas tareas de exploración y pruebas que tiene que desarrollar en su Notebook no se ejecuten de la forma óptima en Databricks y por eso una simple instancia de computación con un JupyterLab sea más que suficiente y Azure ML sea lo más adecuado para su trabajo diario.

Pero ¿qué pasa cuando se cambia de la fase de exploración con un subconjunto pequeño de los datos y se empieza a explorar millones y millones de registros? ¿Qué pasa cuando queremos ejecutar un entrenamiento completo con los datos reales? Azure ML nos ofrece poder crear un cluster de computación o, incluso, poder conectar a un cluster de Databricks, pero recordar que es posible que si estamos en JupyterLab nuestro código no sea el más óptimo para ejecutarse en Databricks.

La unión hace la fuerza

Está claro que no hay ningún servicio que cubra todas nuestras necesidades y siempre vamos a tener que adaptarnos a alguno de ellos, es por eso que, en este caso, lo habitual es hacer uso de Azure Databricks y Azure ML combinados para tener lo mejor de ambos servicios. Azure Databricks como servicio para entrenar nuestros modelos, Azure ML como servicio de exploración y desarrollo de algoritmos, pero también, independientemente de usar o no los Notebooks de Azure ML, se usan los servicios de publicación de modelos de Azure ML desde Databricks. Para cada organización y teniendo en cuenta las capacidades del equipo, las necesidades del algoritmo y los datos se utilizará una combinación de ambas, no existe una combinación única, ideal y perfecta para todo.

ALBERTO DIAZ MARTIN

Microsoft Regional Director

Microsoft Azure MVP | Office Apps & Services MVP

adiazcan@hotmail.com

@adiazcan

<http://blogs.encamina.com/por-una-nube-sostenible/>



32

Asignación de políticas en Microsoft Teams

Hace poco he tenido que preparar un tenant para el correcto despliegue de Teams en una compañía con más de 800 usuarios. Su idea era activar o desactivar ciertas funcionalidades de Microsoft Teams en función del perfil del usuario. Como tenían tres perfiles diferentes, lo que hicimos es crear tres políticas diferentes en cada uno de los apartados del centro de administración de Teams y establecer en cada política las funcionalidades a las que podría acceder el usuario.

"Su idea era activar o desactivar ciertas funcionalidades de Microsoft Teams en función del perfil del usuario"

Para facilitar la administración de estas políticas, se crearon tres grupos en el Directorio Activo (DA) de la compañía que contenían los usuarios de cada uno de los tres perfiles.

Actualmente dentro de Teams hay una serie de políticas que se pueden asignar directamente a un grupo de Microsoft 365. Cuando accedamos a ellas, se mostrará la opción de 'Group policy assignment' (o 'Asignación de directivas de grupo' si lo visualizas en español):

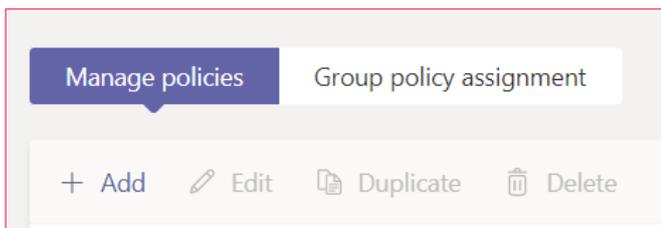


Imagen 1.- Acceso a la asignación de política de grupo.

Al acceder a esa pestaña, lo único que hay que hacer es añadir el grupo y asignarle la política. Se establece también un orden (rank) que indica qué política actuará primero en caso de que el usuario perteneciese a varios grupos.

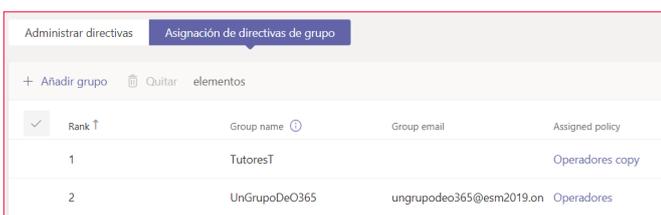


Imagen 2.- Grupos asignados.

Hay que tener en cuenta que las políticas se aplican a los

usuarios siguiendo una jerarquía. Prevalen las que se apliquen a un usuario de forma directa (Direct) y posteriormente las que se asignen a través de grupos (en el orden del ranking). Si el usuario no tiene ninguna política asignada de estos dos tipos, se le aplicará la política Global de la compañía (Org-wide default). Se puede ver en el siguiente esquema:



Imagen 3.- Jerarquía de Políticas.

El problema que me encontré es que, actualmente, hay una serie de políticas en Teams que no se pueden asignar a través de grupos, sino que hay que hacerlo de forma di-

recta. Os pongo un listado de las políticas y si es posible asignarlas a través de grupos, bien sea a través del Centro de Administración de Teams, o con PowerShell:

POLÍTICA	CENTRO ADMIN TEAMS	POWERSHELL
Meeting policy (CsTeamsMeetingPolicy)	SI	SI
Messaging policy (CsTeamsMessagingPolicy)	SI	SI
Live events policy (CsTeamsMeetingBroadcastPolicy)	SI	SI
App permission policy (CsTeamsAppPermissionPolicy)	NO	NO
App setup policy (CsTeamsAppSetupPolicy)	NO	SI
Call park policy (CsTeamsCallParkPolicy)	SI	SI
Calling policy (CsTeamsCallingPolicy)	SI	SI
Caller ID policy (CsCallingLineIdentity)	NO	NO
Teams policy (CsTeamsChannelsPolicy)	SI	SI
Emergency calling policy (CsTeamsEmergencyCallingPolicy)	NO	NO
Emergency call routing policy (CsTeamsEmergencyCallRoutingPolicy)	NO	NO
Dial plan (CsTenantDialPlan)	NO	NO
Voice routing policy (CsVoiceRoutingPolicy)	NO	NO

Para poder hacer la asignación de forma masiva en el caso de las políticas que no se asignan mediante grupos (a través del Centro de Administración), programé un script de PowerShell que me facilitase esa labor. Se trata de sacar los usuarios de uno de esos grupos del M365 y recorrer los usuarios para ir asignándoles las políticas de forma directa.

"dentro de Teams hay una serie de políticas que se pueden asignar directamente a un grupo de Microsoft 365"

Os pongo alguna de las líneas para que podáis utilizarlas si os veis en la misma problemática que yo:

- En primer lugar, cargamos los módulos necesarios:

```
#Instalación del módulo de Teams y Skype
Install-Module -Name MicrosoftTeams
Import-Module SkypeOnlineConnector
```

- En un Array guardamos todas las políticas que se asignan a nivel de Usuario (en mi caso, sólo necesitaba asignar 4 porque en las otras se utilizan las Globales):

```
$DirectPoliciesTypes = @(‘TeamsAppPermissionPolicy’,
‘TeamsAppSetupPolicy’, ‘CallingLineIdentity’, ‘TeamsEmergencyCallingPolicy’)
```

- En otra variable guardo el nombre de la política a aplicar (en mi caso llamé igual a todas):

```
$namePolicy = "NAMEPOLICY"
```

- Ahora hacemos la conexión (en la variable adminUser ha de estar un usuario con permisos de administrador de Microsoft Teams):

```
$creds= Get-Credential -UserName $adminUser -Message
“Credenciales del usuario.”
$connAz = Connect-AzureAD -Credential $creds
$connTeams = Connect-MicrosoftTeams -Credential $creds
$fbSession = New-CsOnlineSession -Credential $creds

Import-PSSession $fbSession
```

- Obtenemos los usuarios (miembros) del grupo que queramos (variable grupoAD):

```
$myGr=Get-AzureADGroup -Filter “DisplayName eq ‘$(grupoAD)’”

$members= Get-AzureADGroupMember -ObjectId $myGr.ObjectId -All 1
```

NOTA: Se incluye -All 1 para que nos devuelva todos los miembros del grupo. Si no se incluye este parámetro, sólo nos devolverá los 100 primeros.

- Ahora recorreremos todos los miembros que tenemos en members y le asociamos la política de forma directa. Esto sólo lo haremos si no la tenía ya aplicada, por eso primero obtendremos qué política tiene y, si no es igual, se la asociamos:

```
$n=0 #Se utilizará para ir contando los miembros del grupo
foreach($m in $members)
{
    $n++

    #Recorremos las políticas que queremos aplicar a cada usuario
    #Políticas de Usuario (Direct)
    Write-Host “POLÍTICAS QUE SE ASIGNAN DIRECTAMENTE AL USUARIO (DIRECT)” -ForegroundColor Cyan
    $nup=0
    foreach($p in $DirectPoliciesTypes)
    {
        $nup++
        Write-Host “$( $nup): Política Direct ‘$( $p)’” -ForegroundColor Gray
        #Vemos qué política tiene el usuario. Si el Get devuelve null, es que el usuario no está en Teams. Si el PolicyName es null, es que tiene asociada la política Global.
        $userPolicies2=Get-CsUserPolicyAssignment -Identity $m.UserPrincipalName -PolicyType $p
        if ($userPolicies2 -ne $null)
        {
            $polname=”Global”
            if ($userPolicies2.PolicyName -ne $null)
            {
                $polname = $userPolicies2.PolicyName
            }
            Write-Host “$( $nup): El usuario $( $m.GivenName) tiene asignada la política ‘$( $polname)’” -ForegroundColor Gray
            if ($userPolicies2.PolicyName -eq $namePolicy)
            {
                Write-Host “$( $n)MIEMBRO$( $m.GivenName)$( $m.UserPrincipalName)DIRECT|El usuario ya tiene la política ‘$( $polname)’ en el tipo ‘$( $p)’” -ForegroundColor DarkYellow
            }else
            {
                Write-Host “$( $n)MIEMBRO$( $m.GivenName)$( $m.UserPrincipalName)DIRECT|Se asigna la política ‘$( $name-
```

```
Policy)' en el tipo '$($Sup)' al usuario." -ForegroundColor White
try{
    switch($Sup)
    {
        'TeamsAppPermissionPolicy' (Grant-CsTeamsAppPermissionPolicy -Identity $m.UserPrincipalName -PolicyName $namePolicy)
        'TeamsAppSetupPolicy' (Grant-CsTeamsAppSetupPolicy -Identity $m.UserPrincipalName -PolicyName $namePolicy)
        'CallingLineIdentity' (Grant-CsCallingLineIdentity -Identity $m.UserPrincipalName -PolicyName $namePolicy)
        'TeamsEmergencyCallingPolicy' (Grant-CsTeamsEmergencyCallingPolicy -Identity $m.UserPrincipalName -PolicyName $nameEmergencyPolicy)
    }
    Write-Host "$($n)|MIEMBRO|$($m.GivenName)|$($m.UserPrincipalName)|DIRECT|Política asociada correctamente" -ForegroundColor Green
} catch {
    Write-Host "$($n)|MIEMBRO|$($m.GivenName)|$($m.UserPrincipalName)|DIRECT|ERROR:'$($_.Exception.Message)': $($_.ScriptStackTrace)" -ForegroundColor "Red"
}

} else {
    Write-Host "$($n)|MIEMBRO|$($m.GivenName)|$($m.UserPrincipalName)|DIRECT|NO SE HAN ENCONTRADO POLITICAS. EL USUARIO NO ESTÁ EN TEAMS" -ForegroundColor Magenta
    break
}
}
}
```

Con este script conseguiréis asociar las políticas Direct a los usuarios de un Grupo de M365.

"hay una serie de políticas en Teams que no se pueden asignar a través de grupos"

Yo tuve algún problema con la conexión del PowerShell ya que al cabo de un tiempo se perdían las credenciales. Lo que hice es meter una validación al principio del foreach de los usuarios para poder continuar a partir del número de usuario en el que se me desconectó (por ejemplo, en el usuario número 284):

```
if ($n -lt 284)
{
    continue
}
```

NOTA: Si lo que queréis es quitar alguna política directa a un usuario para que le aplique la que le correspondiese por Grupo ,o la Global, debéis usar esta instrucción:

```
Grant-CsTeams<nombre política> -Identity $m.UserPrincipalName -PolicyName $null
```

Por ejemplo:

```
Grant-CsTeamsMeetingPolicy -Identity $m.UserPrincipalName -PolicyName $null
```

Pues nada más, espero que os haya resultado interesante y os sirva de ayuda si en algún momento tenéis una problemática similar a la que yo me encontré en este cliente.

ENRIQUE SÁNCHEZ MORENO
Cloud Services Manager en ILUNION IT SERVICES



34

Entrevista Kabel



¿Por qué y cómo empezó en el mundo de la tecnología?

Fundada en 1995, somos una compañía tecnológica española con sede en Madrid y oficinas en Coruña y Granada. Nuestro objetivo es ser el impulsor tecnológico de la transformación de las organizaciones. Para ello ideamos, construimos y acompañamos a nuestros clientes en la creación de soluciones para la revolución digital. Aunque no es nuestro único socio tecnológico, Kabel se ha erigido según los últimos rankings publicados por la Redmond Channel Partner como el Primer Partner español de Microsoft por competencias y el #19 a nivel mundial. Disponemos también de una especialización avanzada en Modernización de Aplicaciones Web sobre Azure y somos "AI inner circle Partner".

Nos gusta que nuestras acciones generen un impacto positivo en nuestros clientes y en la sociedad. Pensamos que la tecnología está cambiando el mundo que nos rodea y tenemos la oportunidad de hacer que ese impacto sea positivo.

¿Cuáles son las principales actividades tecnológicas hoy en día?

Principalmente Kabel presta servicios de asesoría tecnológica a empresas de referencia en sus respectivos sectores. Usamos tecnologías como la Inteligencia artificial para generar algún tipo de ventaja competitiva para nuestros clientes. Entre las soluciones que les aportamos tenemos desde servicios y soluciones alrededor de BigData e IA (como el Voice Analytics Assistant para Power BI o Sketch2code), desarrollo de Aplicaciones Inteligentes, Modelado de Experiencias Interactivas en 3D usando Realidad Aumentada y Mixta (como el caso de 3D Moments de IVI), Construimos Middlewares de Integración y Back-ends de microservicios, soluciones de IoT. Hacemos consultoría para la gestión del cambio sobre tecnologías como Office 365 o la Power Platform de Microsoft, Gestionamos Infraestructuras en Cloud y Centros de Excelencia Cloud (CCoE) y ayudamos a nuestros clientes en la adopción de

la nube y a incorporar prácticas como Agile o DevOps en sus procesos. Por último, ofrecemos servicios de Customer Experience Management para las áreas de Marketing y Canales Digitales. Trabajamos por sector generando aceleradores como las de Industria 4.0, plataformas de servicios de Openbanking o gestión del viaje (journey) del paciente en salud, por mencionar algunas...

¿Cuáles son las principales actividades NO tecnológicas hoy en día?

Uno de los elementos que consideramos clave es la ideación de las soluciones a los problemas de negocio de nuestros clientes. Requiere poner al usuario en el centro y pensar en el problema de fondo antes de intentar dar una solución. Uno de nuestros retos es saber traducir las soluciones tecnológicas a un lenguaje que pueda ser entendido por alguien que no esté familiarizado con el mismo. Buscar siempre cual es el problema de negocio que resolvemos y qué se obtiene de nuestro trabajo como beneficio para el negocio. Esto nos obliga a conocer más acerca de nuestros clientes y de las tendencias de cada sector y estamos trabajando en esa aproximación sectorial que permita dar un uso adecuado de la tecnología para cada uno. No pretendemos industrializar las plataformas, pensamos que cada empresa tiene su especificidad y nos gusta cuidar y potenciar esas especificidades, pero si trabajamos en crear aceleradores y Offerings que permitan hacer más ágiles los proyectos y den valor lo antes posible a nuestros clientes.

Realizamos también actividades de Ocio y deporte, desde las partidas de fútbol en la oficina pasando por el equipo de fútbol o las quedadas virtuales para hablar de distintas temáticas o incluso los concursos de tortilla en la cocina de la oficina. Os aseguro que hay nivel...

¿Cuáles son las actividades que realiza en la comunidad técnica?

Promocionamos que nuestros equipos participen dando ponencias en los distintos eventos de comunidad y con

escuelas y universidades, y patrocinamos en la medida de nuestras posibilidades algunas de ellas. En algunos casos apoyamos iniciativas que han impulsado otras empresas y en otros casos son nuestros consultores los que impulsan alguna como puede ser el capítulo del Global Integration Bootcamp en Madrid. Solemos participar en eventos de comunidad como el Microsoft 365 Saturday, el SQL Bootcamp, Azure Bootcamp o las Dynamics Communities con los Power Platforms Marathons. Hemos participado también en eventos y foros de universidades como la UC3M como Techfest o Tajamar. En lo relativo a comunidades, varias personas de Kabel son miembros de la comunidad ExO (Exponential Organizations) que propone un framework para ayudar a las empresas a convertirse en organizaciones exponenciales a partir de iniciativas apoyadas por tecnologías disruptivas.

Por ejemplo, durante el confinamiento de Covid19 hemos realizado hasta 12 Webinars de distintas temáticas para compartir nuestra experiencia y conocimiento en temas como Industria 4.0, Digitalización de procesos, Asistentes de voz para Power Bi, Servicios cognitivos, etc....

Nos gusta colaborar con otras empresas y participamos en Asociaciones y comunidades empresariales como Exponential Organizations, la IAMCP (Asociación de partners de Microsoft), y la APD, así como en comunidades y foros sectoriales...

No olvidamos nuestra Responsabilidad Social como empresa y en los últimos años hemos trabajado con fundaciones como Prodis para acercar la tecnología a personas con alguna discapacidad intelectual (o mejor dicho, con capacidades diferentes). También hemos colaborado con la fundación Asti en el Robotics Challenge para acercar la tecnología a los niños y a los colegios españoles o STEM Talent Girl para ayudar a las niñas a acercarse a las disci-

plinas STEM desde edades tempranas con master classes y mentoring. Hemos impartido con ellos sesiones y labs de Realidad Aumentada/Mixta e Inteligencia artificial en las que han participado buena parte de nuestros equipos. También hemos trabajado con Fundaciones como Voluntechies para acercar la tecnología a niños que están siendo tratados por cáncer en hospitales de Madrid a los que les hemos acercado un poco de distracción con Mixed Reality durante sus tratamientos en los hospitales. Hemos participado también el university hack de Cajamar como patrocinador y Jurado para impulsar iniciativas y proyectos de BigData en centros universitarios de toda España.

¿Cuál es la visión de futuro en la tecnología de acá a los próximos años?

El ritmo al que están sucediendo los cambios y las disrupciones en los distintos sectores es increíble y lejos de tener pinta de disminuir, se está acelerando aún más. Están acaeciendo en paralelo varias olas de disrupción guiadas por diferentes tecnologías como lo es la IA, la impresión 3D, La bioingeniería, la Realidad Virtual-Aumentada-Mixta, la computación cuántica, Blockchain... capaces cada una de ellas por sí solas de transformar nuestra sociedad y que además se retroalimentan entre sí. En ese escenario, creemos que se abrirán grandes oportunidades y tendremos la posibilidad de aportar nuestro conocimiento y esfuerzo para ayudar en ese camino de transformación a nuestros clientes y generar un impacto positivo en nuestra sociedad. Creemos que tenemos unos retos increíbles por delante de nosotros, que son también una gran oportunidad para todos.

Todos los *secretos* del desarrollo de software para Microsoft Office 365



Un nuevo libro de **Gustavo Velez** que expone cómo desarrollar software para Microsoft Office 365: Exchange, SharePoint, Word, Excel, Teams, etc.

El libro evoluciona con Office 365, de manera que los lectores reciben mensualmente las **últimas actualizaciones** de Microsoft y nuevo contenido.

Encuéntrelo en

<https://guitaca.com>

 **guitaca**
publishers

Integrando Azure Cognitives Services con SharePoint Online

En este artículo particular veremos cómo analizar el contenido de imágenes en una biblioteca de SharePoint aplicando OCR mediante el Servicio Cognitivo "Computer Vision". Todos hemos tenido el problema de querer extraer el texto de una imagen de un pdf o de un texto al que le sacamos una foto desde el celular, por lo que este artículo los puede ayudar a solucionar ese inconveniente en pocos minutos.

"Integración entre SharePoint y los Servicios Cognitivos de Azure a través de Power Automate nos abre nuevas posibilidades de procesar documentos"

Si bien no es complicado tendremos que conocer y utilizar el portal de Azure y saber cuáles son los componentes de Power Automate apropiados.

Al final lograremos un proceso que, al cargar una imagen a una biblioteca de SharePoint, rápidamente le extrae el texto identificado y cargará la información extraída en un elemento de lista con la imagen en adjunto como testigo de lo extraído.

- Primero debemos crear el servicio en el Portal de Azure.
- Ingresamos a nuestra suscripción de Azure y vamos a "Create Resource".

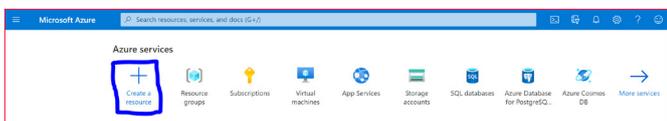


Imagen 1.- Creación del recurso.

- Buscamos en el cuadro de búsqueda "Computer Vision".

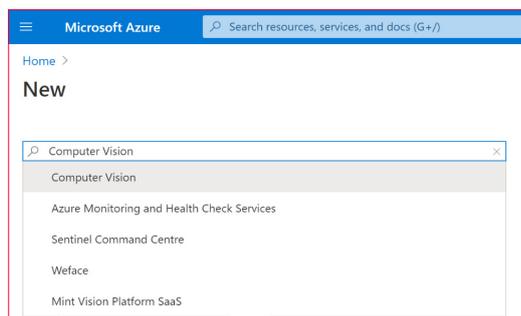


Imagen 2.- Búsqueda del servicio de Computer Vision.

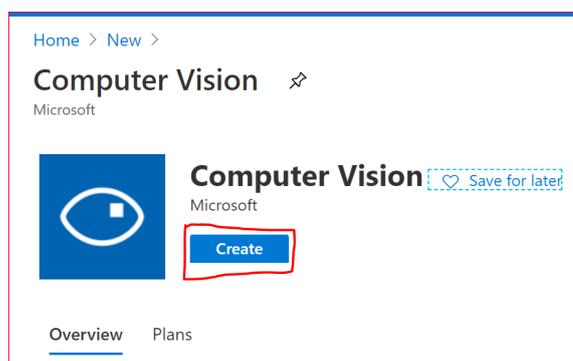


Imagen 3.- Creación del servicio.

- Lo creamos haciendo clic en "Create". En la siguiente pantalla debemos seleccionar una serie de valores.

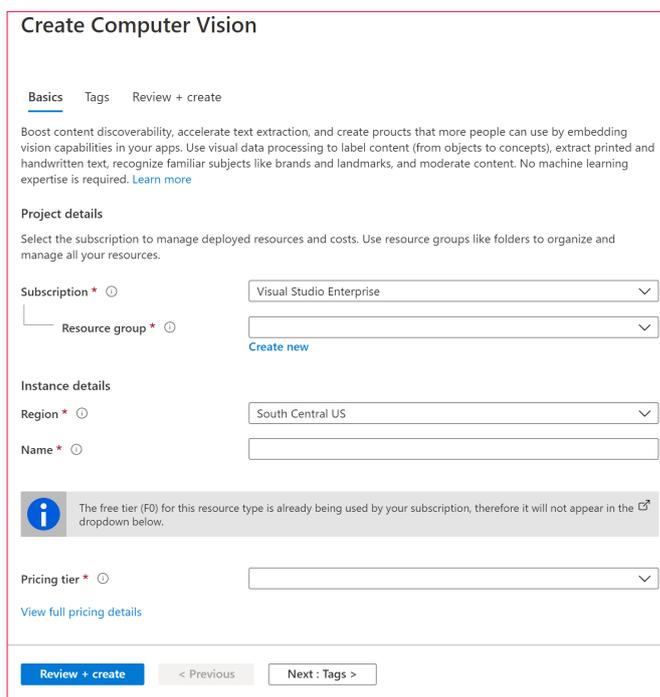


Imagen 4.- Configuraciones del servicio.

- Primero seleccionamos la suscripción y el grupo de recursos en que queremos alojar nuestro nuevo recurso:

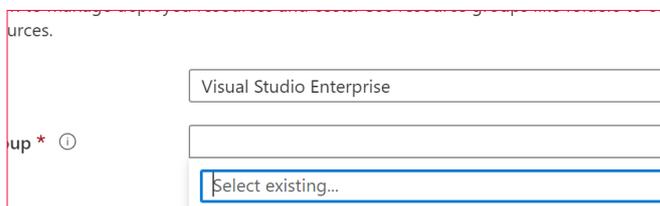


Imagen 5.- Selección de la suscripción.

- Segundo elegimos la región de Azure en la que lo queremos aprovisionar y luego le damos un nombre:

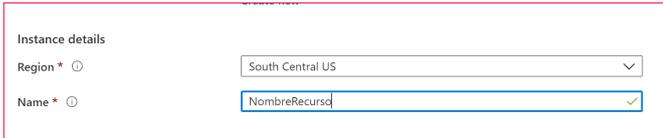


Imagen 6.- Selección de la región de Azure.

(*) Si no está seguro de que región elegir puede ingresar en la siguiente url y elegir la que tenga menor latencia.

- Luego elegimos el "Pricing tier", si usted este experimentando le recomiendo usar el F0 que es gratis y le permitirá probar sin gastar ni preocuparse, luego ya en producción le recomiendo el S1.



Imagen 7.- Selección del Pricing Tier.

- Luego de elegir el precio vamos a Revisar y Crear "Review + create".

(*) Solo puede tener un recurso Computer Vision con Pricing F0 (gratis) por suscripción.

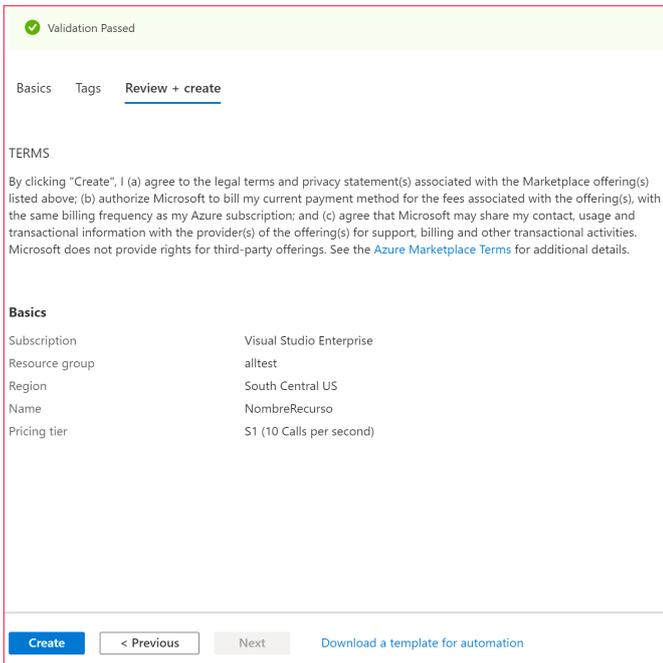


Imagen 8.- Resumen parametrizaciones realizadas.

- Una vez validado le damos "Create".

"explotar el potencial de Microsoft Azure y todos sus servicios en nuestro SharePoint"

Una creado el servicio en Azure debemos crear el flujo en Power Automate que nos procese la imagen y guarde el resultado en un elemento de lista.

- Desde la biblioteca donde se cargarán las imágenes debemos crear el flujo de Power Automate.

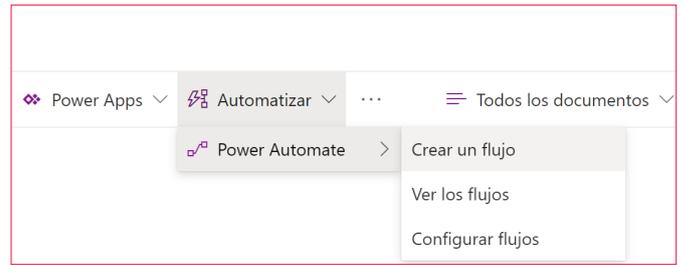


Imagen 9.- Inicio de la creación del Flujo de Power Automate.

- Cuando hacemos clic en Crear un flujo nos aparecerá el siguiente pop up.

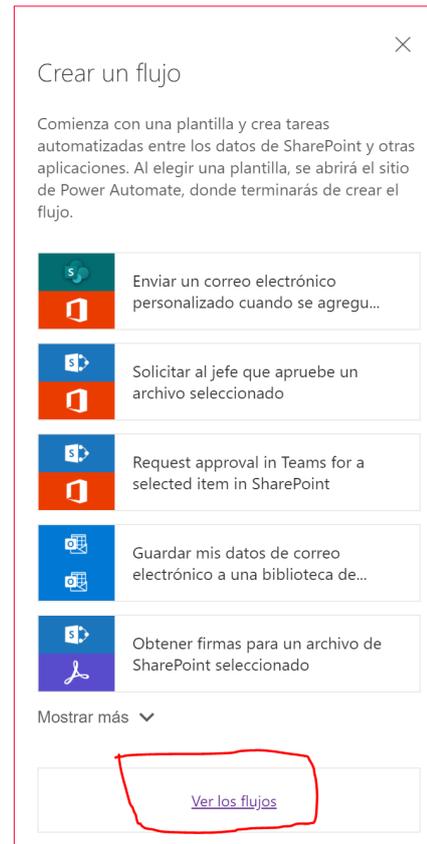


Imagen 10.- Acceso a todos los flujos.

- Hacemos clic en "Ver los flujos" y esto nos lleva al sitio de Power Automate y allí debemos crear un nuevo flujo.

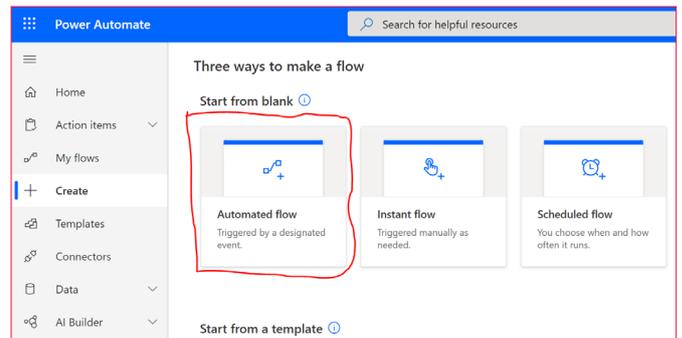


Imagen 11.- Plantilla de inicio para crear el Flujo.

- Luego le damos un nombre y seleccionamos el disparador de tipo "When a file is created in a folder" y lo Creamos.

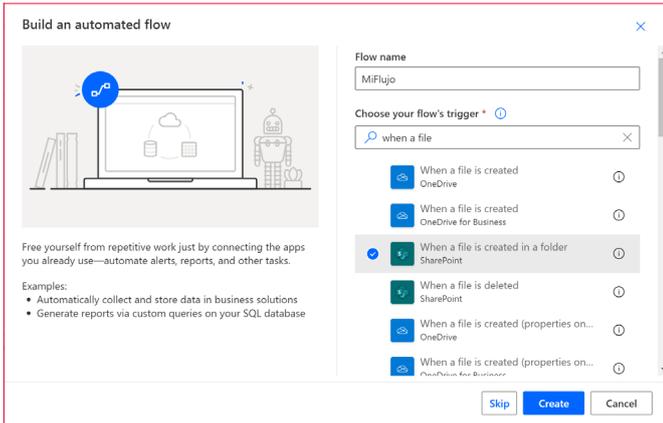


Imagen 12.- Configuraciones iniciales del Flujo.

Una vez en el flujo comenzamos a configurarlo:

- Primero seleccionamos el sitio de SharePoint y la biblioteca a la que le vamos a aplicar el flujo.

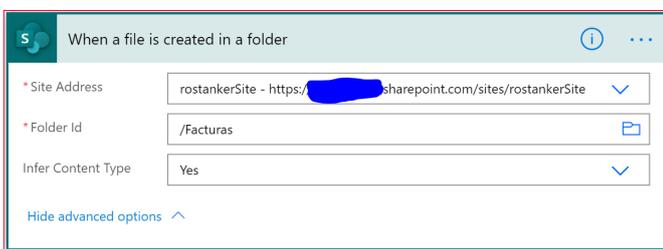


Imagen 13.- Parámetros de configuración de la acción.

- Después agregamos el paso de procesamiento de imagen y lo configuramos:

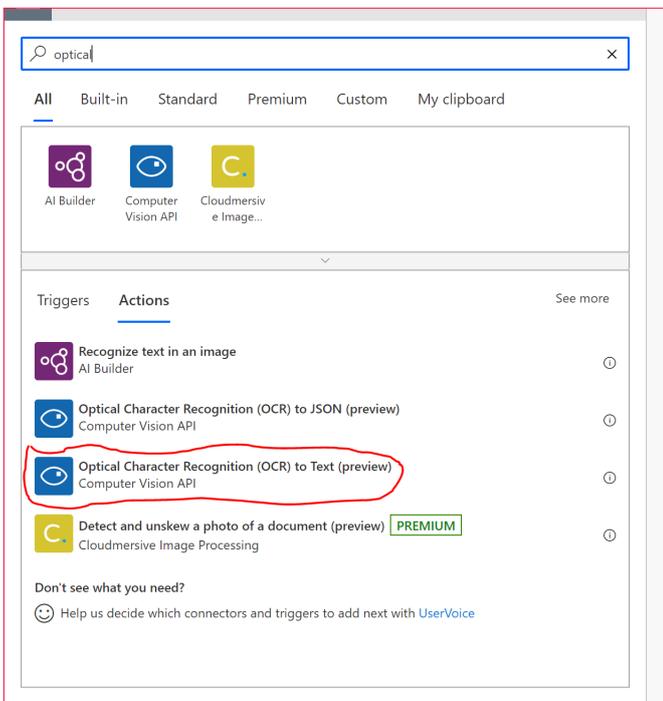


Imagen 14.- Acción para realizar OCR sobre imágenes.

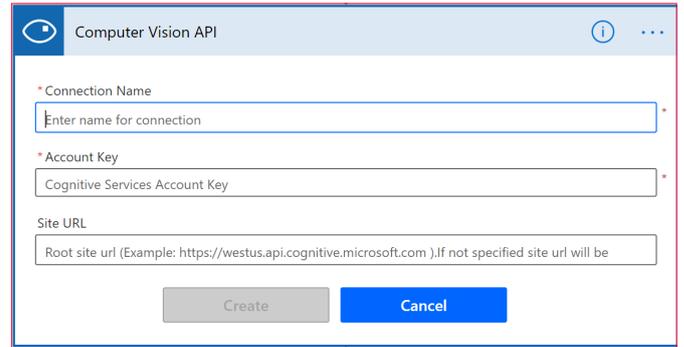


Imagen 15.- Parámetros de configuración de conexión a Computer Vision API.

- En "Connection Name" podemos poner el nombre que más nos guste.
- En Account Key y Site Url debemos ir al recurso en el portal de Azure y obtener los valores de Key1 y EndPoint

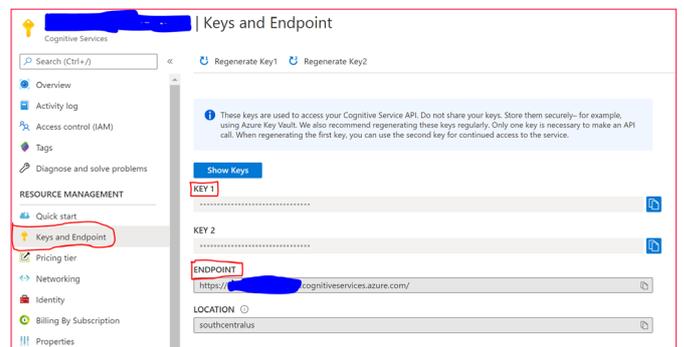


Imagen 16.- Parámetros KEY 1 y ENDPOINT de Computer Vision API.

- Una vez creada la conexión le decimos cual es el archivo que debe procesar:

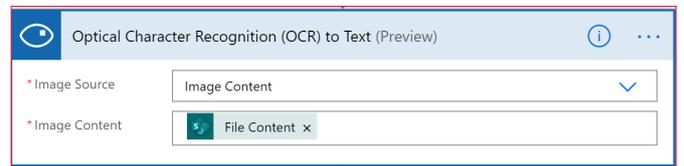


Imagen 17.- Configuración del archivo a procesar.

- Cuando nos posicionamos en Image Content se nos desplegará el contenido dinámico y allí seleccionamos "File Content" que no es más que el contenido del archivo ósea la imagen.

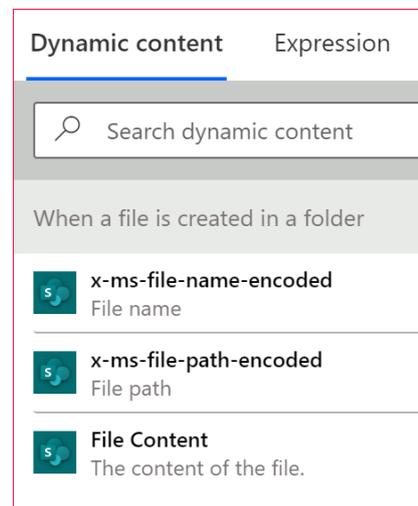


Imagen 18.- Contenido dinámico.

"Una creado el servicio en Azure debemos crear el flujo en Power Automate que nos procese la imagen"

- Paso siguiente inicializamos y cargamos la variable con el resultado del procesamiento:

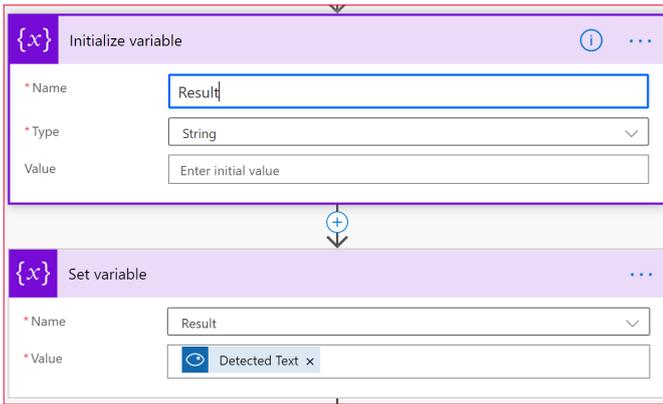


Imagen 19.- Configuraciones de las variables.

- Por último, creamos el elemento de lista de SharePoint y le adjuntamos la imagen.

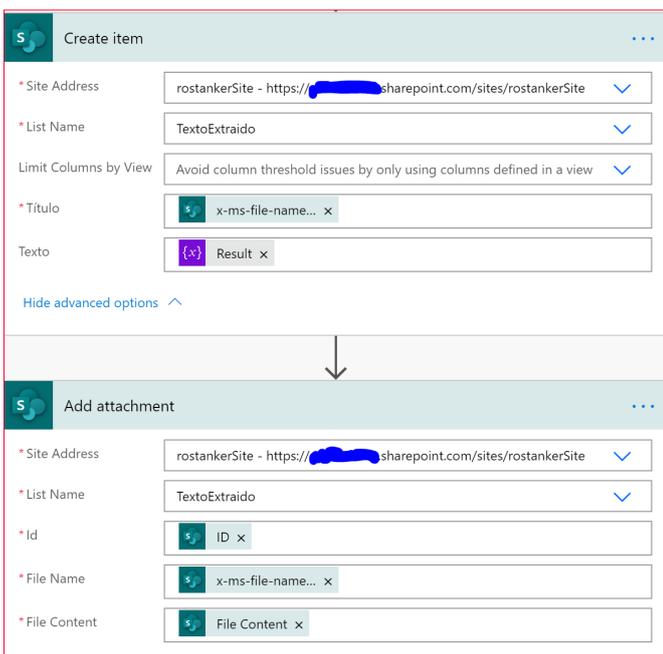


Imagen 20.- Creación del elemento de lista.

Por último, una vez salvado el flujo podemos ir a nuestra biblioteca, cargar una imagen de la que queremos extraer el texto y validar que nuestra solución este correcta y en la lista "TextoExtraido" se esté guardando la información correctamente.

"una vez salvado el flujo podemos ir a nuestra biblioteca, cargar una imagen de la que queremos extraer el texto y validar que nuestra solución este correcta"

Conclusión

Este tipo de integración entre SharePoint y los Servicios Cognitivos de Azure a través de Power Automate nos abre nuevas posibilidades de procesar documentos y extraer información relevante de los mismos sin necesitar escribir una sola línea de código. Esto nos permite explotar el potencial de Microsoft Azure y todos sus servicios en nuestro SharePoint.

ALEX ROSTÁN

Country Manager/ AI & Smarts Applications / Arkano Paraguay

alex.rostan@arkanosoft.com

Twitter: @rostanker

<https://www.linkedin.com/in/alexrostan/>

<http://www.arkanosoft.com>

Powerapps, Azure Maps y Geoespatial Features

En este artículo vamos a revisar la unión de dos Servicios de Microsoft uno del paquete Office 365 y del mundo Power Platform como es PowerApps; y el otro Azure Maps, un servicio de Azure centralizado en el manejo de mapas y todo lo relacionado con geoposicionar datos.

Veremos la nueva versión “Public Preview” de dos componentes que podremos utilizar desde ya en PowerApps con unos sencillos pasos, y todo el potencial de estos si necesitamos jugar con búsqueda de direcciones o posicionar datos en un mapa interactivo dentro de nuestra PowerApps, dejando atrás el control por Imagen que usábamos de forma muy limitada con Google Maps o Bing Maps.

Azure Maps, llegado para quedarse

Cada vez está más asentado este servicio de pago por uso en Azure, y que viene a sustituir al API original de Bing Maps, pasando casi en su totalidad a ser Azure Maps.

Azure Maps para el que no lo conozca es un servicio que ha crecido una barbaridad en los últimos años, no deja de ser un API de servicios Geoespaciales, que podemos instanciar desde nuestras aplicaciones o desde soluciones de Office 365, y en concreto desde el mes de abril en “public preview” también en PowerApps.

El servicio proporciona funcionalidades del tipo:

- Búsqueda de localizaciones: Direcciones, Puntos de interés, coordenadas Geográficas, etc.
- Mapas interactivos: En los que podemos representar todo tipo de datos Geo posicionados, y tenemos componentes en PowerApps o componentes en JavaScript que podemos integrar en nuestras soluciones cliente.
- Rutas y Tráfico: Para montar sistemas de seguimiento GPS para poder hacer por ejemplo tracking de pedidos o rutas de navegación con tu vehículo.
- Sistemas de Geolocalización por IP: Para poder adaptar los contenidos de nuestros sistemas por zona geográfica.
- Servicios meteorológicos: ¿No te quieres mojar? ¿Quieres integrar tu servicio de IoT con el servicio de mapas y actuar en función de la meteorología?
- Servicios de datos: Podemos almacenar datos en el propio servicio, del tipo geoespaciales para evitar latencia en la carga de estos.
- Creator: Permite crear tus propios mapas con aplica-

ciones como AutoCad, y a partir de ellos geoposicionar los datos, hacer servicios de rutas, como por ejemplo digitalizar un mapa de un recinto para un festival de música o poder digitalizar el plano de un hospital.

- Geo-vallas: Marcar una zona de seguridad, y detectar entrada y salida de vehículos, personas o lo que queramos “vigilar” y “controlar”. Muy útil para sistema de conteo de entrada y salida, sistemas de videovigilancia o de seguridad privada..., todo un lujo poder controlar un área si es necesario.

Posiblemente me he dejado alguna característica por evaluar, o nombrar al menos, pero bajo mi modesta opinión están son las principales y las que hacen de Azure Maps un servicio “top” en cuanto al geoposicionamiento de los datos.

Powerapps y Azure Maps , ¿Qué nos aporta esta nueva unión?

Como todos los servicios que se están pasando a Componentes de PowerApps, lo primero que tenemos que entender es que han venido para “facilitarnos” la vida siempre y cuando nos podamos adaptar a la funcionalidad que nos proporcionan.

Es cierto que como con casi cualquier servicio de Azure, ya podíamos instanciar el mismo desde un Componente personalizado (Aplicaciones de Lienzo o Componente en React), haciendo uso de los “Custom Connector” para poder instanciar el API de Azure Maps de forma controlada, y creando una instancia del servicio en Azure.

Todo esto estaba genial, pero requiere “Conocimiento” del API, del modelo de despliegue de los Conectores a medida, y claro está controlar un mínimo Azure y mantener los costes de uso del servicio de Azure Maps de nuestro tenant.

Pues lo dicho, PowerApps viene a ahorrarnos todo esto, y nos proporciona dos componentes que podemos utilizar en fase de pruebas desde ya mismo y que son:

- Mapas Interactivos: Como veremos en los ejemplos, nos permite insertar un mapa 100% interactivos, y posicionar un juego de datos geoposicionados mediante sus coordenadas en el mapa.

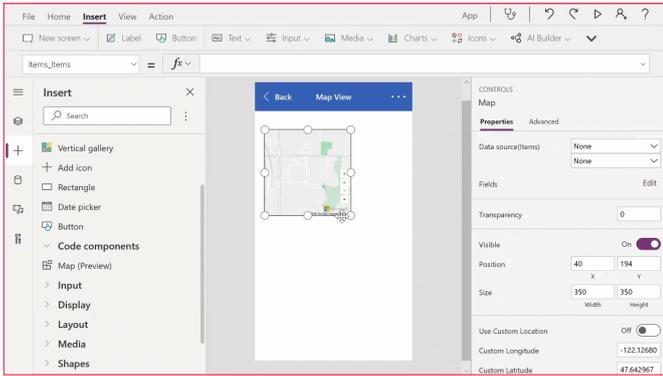


Imagen 1.- Mapa interactivo.

- Sugerencia y búsqueda de direcciones: Este segundo componente nos permite atacar al API de Azure Maps, desde una simple caja de texto y buscar direcciones. Como es de esperar en este tipo de componentes el propio servicio sugiere direcciones una vez empezamos a escribir, y nos corrige si hemos metido algún error en la dirección.



Imagen 2.- Barra de búsqueda de direcciones.

Como empezar a usar estos dos nuevos componentes, configurar el entorno

Si os habéis emocionado rápido, y habéis ido a buscarlo en el catálogo de componente de vuestro PowerApps studio, y no está, podéis estar tranquilos es normal porque debemos habilitar una serie de características a nivel de tenant, y posteriormente en la propia aplicación que estéis editando.

"Azure Maps un servicio "top" en cuanto al geoposicionamiento de los datos"

Lo primero que debemos hacer es acceder a la dirección <https://admin.powerplatform.microsoft.com/> con un usuario administrador y seleccionar el "entorno" en el que vamos a utilizar las nuevas capacidades de Azure Maps. Una vez dentro del entorno debemos darle a "configuración", y en el menú que se nos presenta seleccionar Productos->Características.



Imagen 3.- Configurando el entorno.

Una vez en características debemos habilitar los servicios "Geoespaciales" que como veremos están en versión "pre-view".

Servicios geoespaciales (versión preliminar)

Las características geoespaciales se habilitan al aceptar las **condiciones de servicio**

Activado

Imagen 4.- Habilitar la característica en entorno.

Una vez configurado el entorno, para poder utilizar los componentes de Maps en nuestra App debemos acceder a la aplicación en modo "edición" desde el PowerApp studio, y en el apartado de "configuración->Configuración avanzada", seleccionar en el apartado de "Características experimentales" la opción:

Características geoespaciales [?]

Habilita un componente de mapa interactivo, una entrada de dirección autocompletada y capacidades de conversión de geocodificación. Estas características se deben habilitar primero en el Centro de administración de Power Platform.

Activar

Imagen 5.- Habilitar maps en la app.

Construyendo nuestra primera pantalla con Maps: Añadiendo una posición al mapa

Vamos a crearnos una nueva pantalla en una aplicación de Lienzo en PowerApps, que nos permita añadir un mapa interactivo y sobre este pintar unas coordenadas que nosotros recojamos por pantalla tal y como se ven en la siguiente imagen:



Imagen 6.- Demo 1 añadir posiciones.

Voy a saltarme un poco la parte de diseño, dejando que cada uno le da la forma que quiera, pero en resumen vamos a tener:

- Dos cajas de entrada de texto que podamos escribir Latitud y Longitud a pintar en el mapa.
- Un control booleano que nos permita elegir si queremos que nos geoposicione nuestra posición actual o no. Si elegimos la posición GPS o actual que nos da PowerApps las cajas de texto se van a sobrescribir con las coordenadas exactas.
- Un botón visualizar en mapa que lanzara la posición con “latitud y longitud” al mapa.
- Mapa interactivo que permita renderizar la posición deseada en el mapa.

Añadiendo la recogida de coordenadas

Para ello en el menú insertar vamos a añadir dos componentes del tipo “Entrada de texto” y unos label para poder visualizar “Longitud y Latitud” tal cual vemos en la imagen.

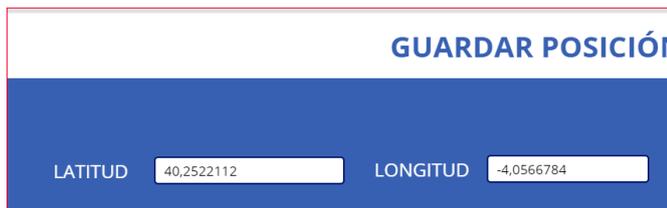


Imagen 7.- Inputs de coordenadas.

Una vez insertados y posicionados a nivel de diseño como nosotros queramos, yo recomiendo ponerles un nombre que nos permita luego buscarlos en la caja de fórmulas de forma sencilla, en mi caso los he llamado text_lon y text_lat.

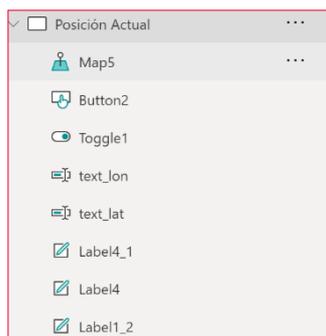


Imagen 8.- Listado de controles.

Ahora nos toca añadir tanto el control booleano como el botón que nos va a permitir almacenar la “posición a renderizar” en el mapa. El control booleano se denomina “alternar” en el menú de controles de entrada, y el otro es un control de tipo botón sencillo, que debemos añadir y dejar como en la primera imagen de la aplicación que ya hemos visto.



Imagen 9.- Botonera completa.

En este punto debemos insertar un par de fórmulas:

- 1.- Seleccionamos el control “si o no” de Posición GPS y añadimos las siguientes fórmulas a los eventos OnCheck y UnCheck tal cual vemos en las siguientes imágenes:

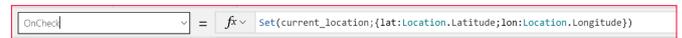


Imagen 10.- Formula oncheck.



Imagen 11.- Formula Uncheck.

Es muy simple de entender, vamos a guardar en una variable “Current_location” los datos que nos proporciona nuestra aplicación de la posición exacta en la que se encuentra, bien sea desde el navegador web vía IP o desde el móvil con la función Posicionamiento.

```
Set(current_location;{lat:Location.Latitude;lon:Location.Longitude})
```

- 2.- Para pasar a los campos de texto la “lat y lon” correctas debemos seleccionarlos y en la propiedad “default”, añadir current_location.lat o current_location.lon.

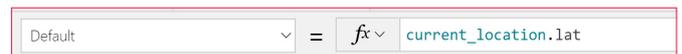


Imagen 12.- Valor de los inputs en default.

- 3.- Por último para terminar con la botonera, debemos configurar el evento OnSelect del botón “Visualizar en mapa” con la fórmula:

```
ClearCollect(Positions;{lat:Value(text_lat.Text);lon:Value(-text_lon.Text);Name:"Current_Position"})
```

Lo que hacemos con esta instrucción es crear una colección de datos Position, en la que guardamos:

- a. Lat (latitud).
- b. Lon (longitud).
- c. Name (el texto Current_Position).

Esto genera una colección de un elemento que vamos a utilizar desde el mapa interactivo, es importante usar una Colección y no una variable, ya que el mapa espera una entrada de datos del tipo tabla, y en caso de usar variable nos va a dar un error en la fase de mapeo.

Añadiendo el componente de mapa Interactivo

Una vez tenemos todos los datos necesarios en nuestra aplicación, podemos pasar a renderizar las posiciones en el mapa. Para ello en el menú insertar, buscamos “Map”, y veremos que ahora una vez habilitado todas las características de geoposicionamiento podemos agregarlo a nuestro lienzo.

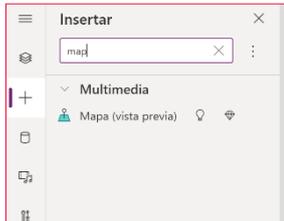


Imagen 13.- Insertando el mapa interactivo.

Como ves en la imagen, Microsoft ya nos deja alguna pista de que este conector va a ser Premium y por eso no nos pide en ningún momento que debemos de alta una conexión a Azure Maps como si nos pasa con otros servicios. El propio componente nos deja conectado al api y nos preocupamos de todo.

El componente es muy sencillo de configurar lo insertamos debajo de nuestra botonera, y lo adaptamos a la pantalla para que ocupe lo que deseemos. Ahora solo nos queda, seleccionar el componente de mapa, y configurar el Origen de datos que será nuestra colección “Positions”, y configurar los campos ItemsLabels, ItemsLongitudes y ItemsLatitudes, tal cual se ve en las siguientes imágenes:

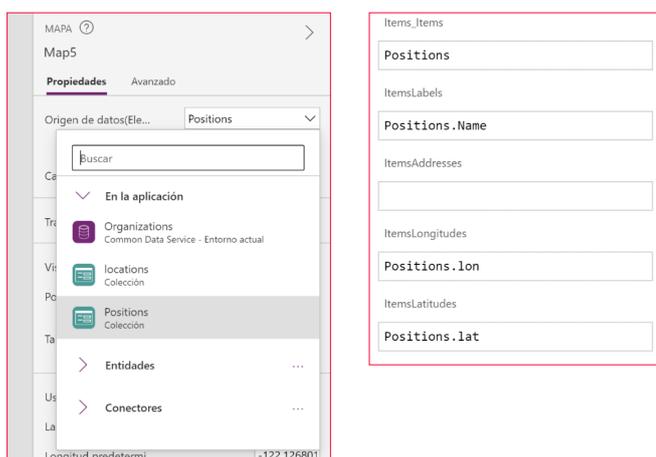


Imagen 14.- Configuración del componente.

Las propiedades que hemos configurado permiten poner como nombre encima de la chincheta el campo Positions. Name que como dijimos antes será CurrentPosition, y por otro lado los campos latitud y longitud que también tiene la estructura de datos definida en Positions. En este caso como en Positions solo estamos metiendo un elemento, el mapa pinta uno, pero como veremos en el siguiente ejemplo el propio control pinta tantos como haya en la colección de datos.

Y con todo esto tenemos un mapa interactivo, que pinta una posición geoposicionada por latitud y longitud.



Imagen 15.- Demo una funcionando.

Por terminar de cerrar la revisión del componente “Mapa interactivo”, podremos ver en el menú del componente que podemos visualizar en el mapa la “ubicación actual” de forma nativa sin hacer la botonera, aunque yo quería ver cómo hacer uso de la función “Location”.

O por otro lado podemos habilitar opciones del mapa como, si queremos que el usuario pueda hacer zoom, inclinar el mapa o ver una vista satélite en vez de relieve. Todas estas opciones dan mucha libertad al usuario de visualizar de la mejor forma posible la posición deseada.

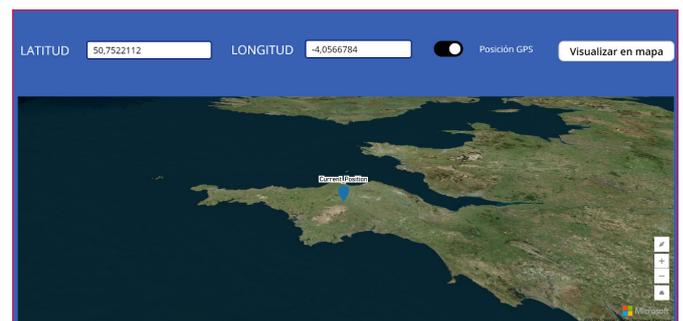


Imagen 16.- Configurando el componente.

Busqueda de una dirección y renderizado de más de un punto en el mapa

Con este segundo ejemplo vamos a ver cómo hacer uso del segundo componente que se llama “Adress suggestion” y poder añadir al mapa tantos puntos como deseemos. La idea del mapa interactivo no va a variar, vamos a escoger una colección de puntos a visualizar y este de forma nativa los va a renderizar. Donde sí debemos poner más atención es en el componente de búsqueda de direcciones que como vamos a ver es muy muy sencillo de utilizar.

“PowerApps es un buen producto, al que le queda un pequeño paso para consolidarse”

Esta nueva pantalla va a permitir añadir desde un campo de búsqueda de direcciones “N” puntos a nuestro mapa interactivo.



Imagen 17.- Añadiendo múltiples posiciones desde una dirección.

La pantalla va a ser igual de diseño libre, pero debe tener los siguientes elementos:

- Un componente de búsqueda de direcciones de “Azure Maps”.
- Un botón Añadir: Para ir metiendo en una colección “Locations” las posiciones.
- Un botón borrar: Vacía la colección “Locations”.
- Un mapa interactivo para renderizar los puntos.

Primero recogemos la dirección a renderizar, insertamos el componente de búsqueda de direcciones desde el menú de componentes, buscándolo por “Entrada de dirección”, tal cual se ve en la siguiente imagen.

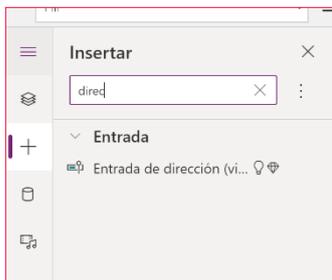


Imagen 18.- insertando barra de búsqueda.

Una vez añadido no tenemos que hacer nada con él, más que configurar un poco el componente, tal cual se ve en la siguiente imagen:

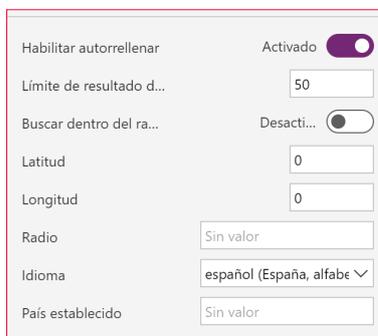


Imagen 19.- Configurando barra de búsqueda.

Como veis se puede limitar el “rango de resultados” que en nuestro caso lo hemos dejado en 50, y no queremos limitar el resultado a un radio de coordenadas como podría ser la Comunidad de Madrid, sino que lo vamos a dejar abierto a cualquier dirección del planeta para aver el rendimiento del componente. Sí vamos a predefinir el idioma a castellano,

para que nos sea sencillo escribir la dirección y nos corrija palabras como Calle, Avenida o Paseo, si lo metemos mal.

Insertar los puntos recogidos en el buscador

Para poder mandar las direcciones encontradas en el buscador en el botón “Añadir” vamos a configurar la siguiente formula en el evento “OnSelect”:

```
Collect(locations;(Name:AddressInput2.UserInput;Lat:AddressInput2.SelectedLatitude;Lon:AddressInput2.SelectedLongitude;City:AddressInput2.Municipality))
```

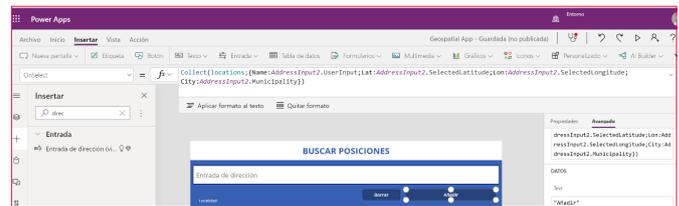


Imagen 20.- Formulando la colección de locations.

Esta fórmula nos permite añadir elementos a la colección “Locations”, desde el buscador de direcciones que en nuestro caso se llama AddressInput2 y cogemos los siguientes campos:

- UserInput: Cadena de texto con lo que hemos escrito en pantalla.
- SelectedLatitude: Latitud de la dirección encontrada.
- SelectedLongitude: Longitud de la dirección encontrada.
- Municipality: Localidad o población de la dirección escogida.

En el botón borrar simplemente vamos a añadir en el evento OnSelect la fórmula Clear(locations) para poder borrar la colección de datos Locations.

Renderizando y probando nuestra app

Vamos a insertar un mapa interactivo, y añadir como origen de datos la colección Locations (tal y como hicimos en el ejemplo anterior), y seleccionaremos los siguientes campos para la leyenda y la posición:

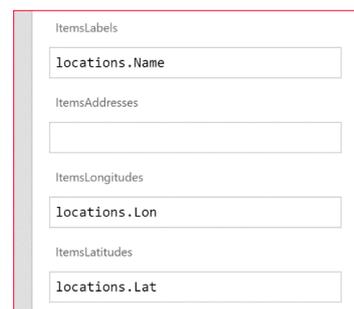


Imagen 21.- Configurando el mapa.

Ahora podríamos lanzar la aplicación, añadir varias direcciones de Madrid como podría ser Calle Gran Vía 20 o Calle Alcalá 200:



Imagen 22.- Buscando direcciones.

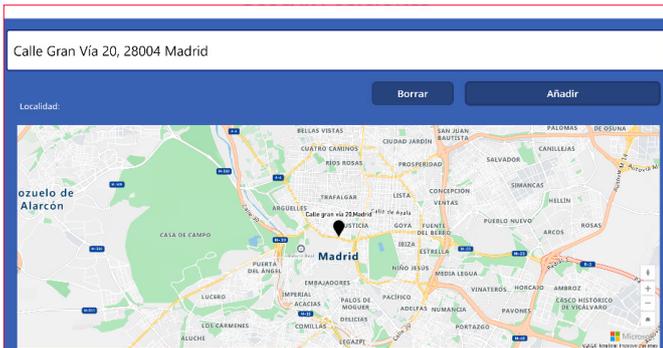


Imagen 23.- Renderizando direcciones.

Si agregamos varios puntos relativamente cerca, según vayamos alejando el Zoom veremos que se nos agrupan con un contador sobre la propia leyenda, haciendo muy fácil localizar el número de puntos a posicionar.

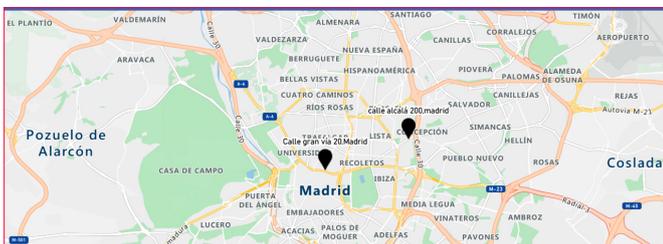


Imagen 24.- Varios puntos.



Imagen 25.- Agrupando puntos.

Interactuando con el mapa

Una cosa que aún no hemos visto es que el mapa carga una serie de datos, pero también expone una serie de eventos para poder interactuar con él, y hacerlo muy integrable en la pantalla. Por ejemplo, podemos utilizar el evento de clic de una chincheta sobre el mapa, para que por ejemplo nos pinte una pequeña leyenda. En este caso podemos añadir un label encima del mapa para poder pintar la “localidad” a la que hace referencia ese punto exacto del mapa.

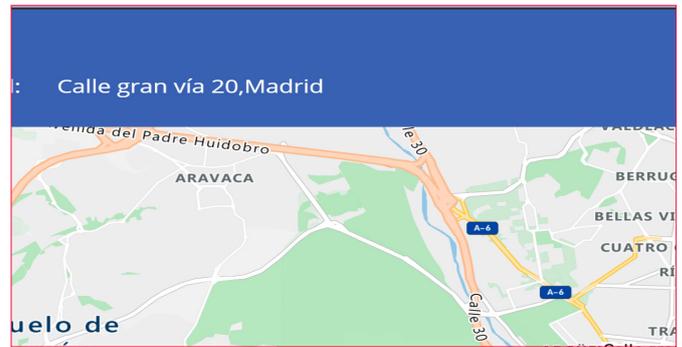


Imagen 26.- Interactuando con el mapa.

Para poder pintar la localidad es tan fácil como en el control label, en el campo “Text” añadir `First(Map2.Items_SelectedItems).Name`, donde “Map2” es el mapa y “Items_SelectedItems” son los puntos del mapa seleccionado. En nuestro caso utilizamos la función “First” para coger solamente un punto, que en este caso es el último que se ha seleccionado ya que están ordenados por antigüedad.

Un servicio más a PowerApps, más leña al fuego

Bueno una funcionalidad más, para un servicio de PowerApps ya de por sí potente, pero al que aún le quedan cosas serias por pulir como es la experiencia de trabajar con componentes, poder tener un entorno de trabajo completo como puede ser Visual Studio Code, el modelo de despliegue y la configuración de las conexiones entre entornos.

Es cierto que nos encontramos lo que esperamos, un componente de Mapas muy sencillo de usar, potente, aunque con muchas características por añadir como son las rutas dentro de un mapa o añadir Shapes o áreas en el mismo.

Sin duda añade un buscador muy potente, que aun sin limitar el rango de resultados tiene un rendimiento muy notable; y el mapa interactivo permite interactuar con el mapa y hacer uso de los datos del api de forma muy ágil y directa sin tener necesidad de conocer el contrato del api de Azure Maps más que lo necesario.

Seguimos echando “leña al fuego”, ahora nos queda ver si el “Guiso” nos sale bueno y podemos hacer aplicaciones cada vez más complejas y con equipos grandes de trabajo, o se va a quedar en una herramienta para hacer pequeños componentes muy puntuales en nuestras aplicaciones móviles, o del entorno Office 365 como son SharePoint o Dynamics 365.

El tiempo lo dirá, es cierto que ahora mismo PowerApps es un buen producto, al que le queda un pequeño paso para consolidarse entre las comunidades de desarrollo de aplicaciones del mundo Microsoft, ¡¡aunque todo es posible!!

SERGIO HERNÁNDEZ MANCEBO
Principal Team Leader
Azure MVP
@shmancebo

Introducción a Azure SQL

En este artículo nos enfocaremos en las diferentes presentaciones del motor de bases de datos relacional SQL en la nube de Microsoft Azure. Conoceremos las diferentes presentaciones enumerando las características y beneficios de cada una de ellas.

La familia de SQL Cloud database nos ofrece varias posibilidades para poder migrar nuestras aplicaciones a la nube de Microsoft modernizando nuestro entorno y ofreciendo diferentes opciones para los desarrolladores.



Comencemos mencionando que la familia de Azure SQL tiene sus orígenes en el superconocido Microsoft SQL Server, con muchos puntos en común.

"es muy fácil escalar el servicio para tener más recursos"

Ya desde hace algún tiempo Microsoft se enfoca en desarrollar nuevas funcionalidades primero en la nube para luego llevarlas al producto SQL Server, por lo que la mayoría de estas tienen muchos puntos en común. Como ejemplo, las relacionadas a mejoras en performance como "Query Store", "Automatic Query Tuning" e "Intelligent Query Processing" fueron testeadas e implementadas primero de esta forma.

Componentes de Azure SQL

Son 3 las diferentes opciones de Azure SQL.

- **SQL Server en Azure Virtual Machines:** Microsoft nos da la posibilidad de crear Virtual Machines en Azure corriendo la misma versión de SQL Server que utilizamos On-Premise, pudiendo seleccionar y combinar entre una enorme cantidad de Sistemas Operativos

(Incluyendo varias distribuciones de Linux), versiones y ediciones de SQL Server.

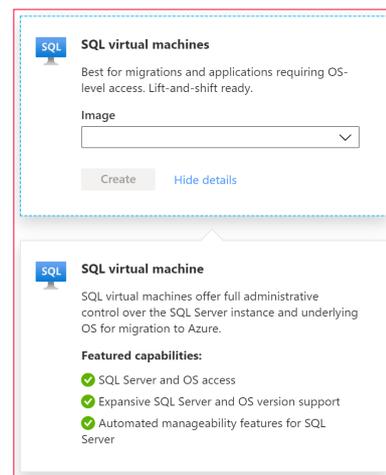


Imagen 1.- SQL Server Virtual Machines en Azure.

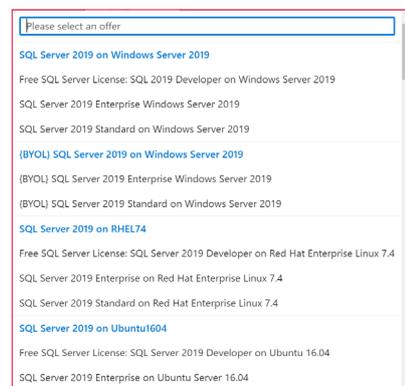


Imagen 2.- Versiones de SQL Server disponibles para VMs en Azure.

Esta alternativa que se encuentra dentro de las opciones de Infraestructura como Servicio (IaaS) nos da la posibilidad de muy rápidamente poder migrar nuestras bases de datos que están en servidores de nuestra organización a Azure, sin necesidad de realizar testing exhaustivo dado que es la misma versión de SQL Server que ya venimos utilizando y conocemos. Hay algunos puntos que debemos poner principal atención dado que hay tener en cuenta que debemos encargarnos de toda la infraestructura para soportar SQL Server, esto es la configuración de networking, la autenticación con Azure Active Directory, esquemas de backup, alta disponibilidad y disaster recovery. También es importante mencionar que nosotros debemos encargarnos de aplicar las actualizaciones a Windows y a SQL Server, aunque en los últimos cambios se permite delegar esto

último a Microsoft

- **Azure SQL Database:** Esta opción, que es la primera disponible en la forma de Plataforma como servicio (PaaS) y que está disponible desde los inicios de Azure (con muchísimas mejoras desde las primeras presentaciones) nos permite delegar muchas de las tareas de administración (HA/DR, Backups, Autotuning, etc.) dado que es SQL Server como servicio manejado pero con una característica muy particular.

Podemos tener una única base de datos aislada, por lo que todos los objetos de nuestra aplicación deben estar contenidas dentro de esta única base. Por ejemplo, una Stored Procedure solo puede acceder a tablas y vistas dentro de esa misma base de datos. Esto nos permite utilizar este servicio para crear aplicaciones modernas “Cloud” que se basan en una única fuente de datos.

Otra de las características importantes es que muy fácilmente podemos escalar el servicio para tener más recursos. Esto lo podemos realizar manualmente, en horarios particulares (Schedule) o bajo reglas de patrones de ubicación del servicio.

Otra referencia a la forma de escalar son los distintos niveles de servicios (Service Tiers) como General Purpose o Business Critical con opciones de DTU o vCore (Virtual Cores).

También con opciones de “Serverless” que nos permite “pausar” el servicio si no se detecta utilización de este y también la opción de “Hyperscale” que nos da la posibilidad de tener bases de datos de más de 100 TB. Todo esto con niveles de servicio (SLA) de 99.99%

- **Azure SQL Database Managed Instance:** Esta opción de SQL Server como servicio manejado es la más reciente de todas y nos permite tener una compatibilidad de casi 100% con las funcionalidades del producto SQL Server. Es básicamente el mismo SQL Server que utilizamos on-premise, incluyendo el SQL Server Agent, replicación transaccional, service broker, etc. Con el agregado importantísimo de las características de un servicio PaaS, donde los backups, la aplicación de actualizaciones y la configuración de alta disponibilidad forman parte de la oferta, por lo que el usuario no debe encargarse de ellos.

Migrar desde SQL Server on-premise a Azure SQL Database Managed Instance (MI) es muy sencillo, solamente realizar el backup de nuestras bases de datos a un BLOB Storage para luego poder restaurarlas en nuestra MI por lo que mover nuestra aplicación a este servicio solo requiere que modifiquemos el string de conexión.

Debemos considerar que las funcionalidades de FileStream y Filetable así como Transacciones Distribuidas no están disponibles en Managed Instance.

Migrando hacia Azure SQL

Microsoft nos ofrece la posibilidad de realizar la migración de nuestras bases de datos locales utilizando la aplicación “Data Migration Assistant” (DMA) que nos permite llevar a cabo un “Assesment” o evaluación de las bases de origen, pudiendo detectar problemas que puedan afectar la migración. Asimismo, nos genera un reporte detallado indicando como resolver estos inconvenientes

Cuando seleccionamos Azure SQL Database como destino para nuestra migración, DMA puede detectar funcionalidades parcialmente compatibles, recomendando enfoques alternativos disponibles en Azure.

Para migraciones de gran escala, donde tengamos que mover muchas bases de datos de gran tamaño, Microsoft recomienda utilizar “Azure Database Migration Service”

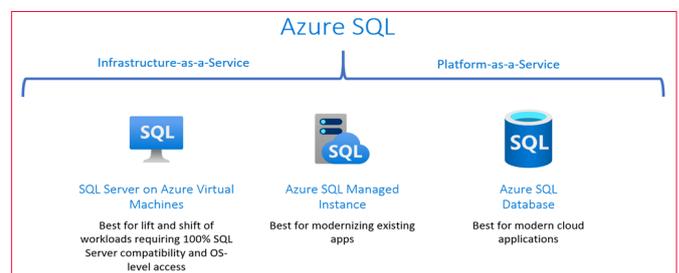


Imagen 3.- Versiones de Azure SQL.

Conclusión

La familia de Azure SQL nos da la posibilidad de seleccionar el destino que mejor se adecue a nuestras necesidades y nos da las herramientas para llevar a cabo la migración a una plataforma de datos moderna.

Un camino posible para realizar nuestra migración, para el caso donde tengamos una aplicación de misión crítica que llevemos utilizando por varios años, es llevar nuestro SQL Server Local a una Máquina Virtual corriendo SQL Server 2017 o 2019, manteniendo el nivel de compatibilidad de las bases de datos de usuario tal cual como en el servidor de origen. Esto nos permitirá tener presencia en Azure muy rápidamente, manteniendo los mismos niveles de performance. Como paso adicional lo ideal es llevar esas bases a “Azure SQL Database Managed Instance” para así poder tener nuestra bases y aplicaciones en una plataforma moderna, y todo esto sin con mínimo esfuerzo.

Para el caso en donde tengamos que desarrollar una aplicación nueva, la recomendación es considerar Azure SQL Database como la primera opción.

JAVIER VILLEGAS

Global Database Administrator & Design Coordinator @ MSC

Microsoft MVP , Data Platform

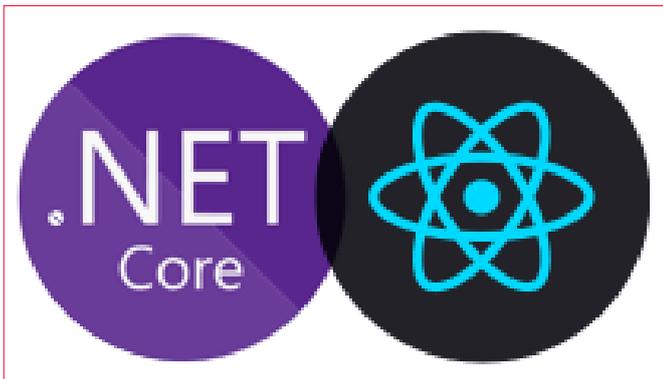
javier.ignacio.villegas@gmail.com

@javier_vill

/javiervillegas

ASP.Net Core con ReactJS

.Net Core: un nuevo framework escrito desde cero, gratuito, de código abierto y sucesor multiplataforma de .Net Framework.



No cabe a duda, que hoy ReactJS es una de las librerías más usadas para desarrollo de aplicaciones web de una sola página y que ASP.Net Core es también un framework que día a día gana más mercado, principalmente en el desarrollo de REST APIs. Por tanto, la combinación de ReactJS con ASP.Net Core Web API es algo que Microsoft ha pensado muy bien, de modo que podamos crear de forma muy simple, desde la comodidad del Visual Studio, un proyecto de ASP.Net Core que nos brinde una REST API y que sirva como frontend un proyecto React.

"ReactJS es una de las librerías más usadas para desarrollo de aplicaciones web"

Pasos a seguir

- 1.- Lo primero que haríamos sería crear un nuevo proyecto de Aplicación Web con ASP.Net Core.
- 2.- Una vez que nos muestra las plantillas a seleccionar para nuestro proyecto seleccionamos React.Js.

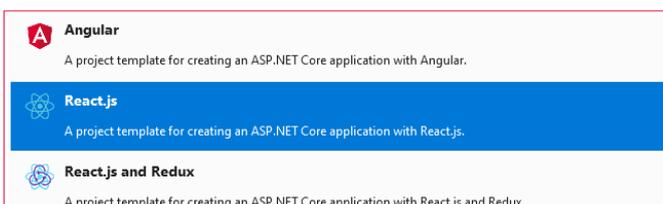


Imagen 1.- Plantilla a seleccionar.

- 3.- Una vez creado el proyecto podemos observar en el Explorador de soluciones nuestro proyecto.

En la carpeta ClientApp, tendremos nuestro proyecto ReactJS ya creado. Al ejecutar nuestro proyecto el Visual Studio por sí solo, descargará las dependencias de NodeJS para nuestro proyecto ReactJS.

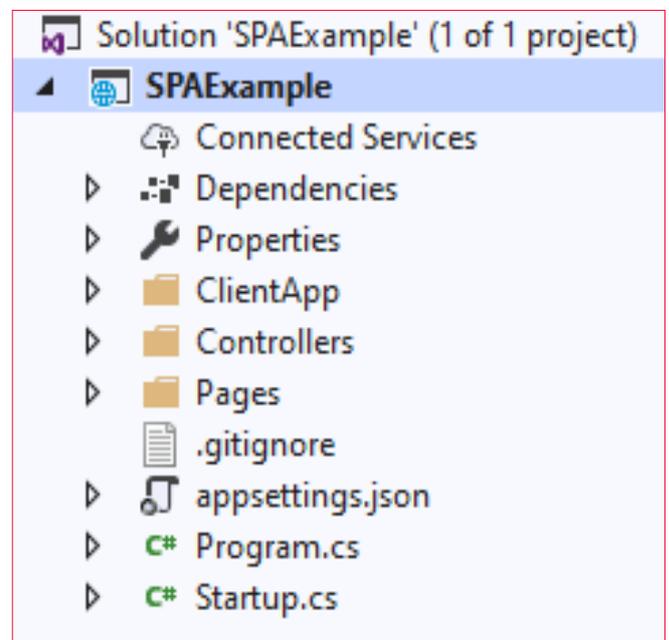


Imagen 2.- Estructura del Proyecto en Visual Studio.

Entonces, un análisis rápido del archivo Startup.cs, nos permitirá distinguir el código específico que garantiza la integración con ReactJS.

```

0 references | 0 exceptions
public void ConfigureServices(IServiceCollection services)
{
    services.AddMvc().SetCompatibilityVersion(CompatibilityVersion.Version_2_1);

    // In production, the React files will be served from this directory
    services.AddSpaStaticFiles(configuration =>
    {
        configuration.RootPath = "ClientApp/build";
    });
}

```

Acá podemos observar cómo se añade la carpeta donde una vez hecho publish estarán los archivos para producción. Por defecto, ReactJS crea los archivos para producción en la subcarpeta /build dentro de la raíz del proyecto. Los archivos dentro de esta carpeta se usarán solamente si el proyecto ASP.Net Core esta publicado en producción.

```
app.UseSpa(spa =>
{
    spa.Options.SourcePath = "ClientApp";

    if (env.IsDevelopment())
    {
        spa.UseReactDevelopmentServer(npmScript: "start");
    }
});
```

"un análisis rápido del archivo Startup.cs, nos permitirá distinguir el código específico que garantiza la integración con ReactJS"

A continuación, podemos ver como establecemos que la ruta del código fuente de nuestro proyecto. Y establecemos, además que, si el proyecto ASP.Net Core está corrien-

do en desarrollo, usa el proyecto React como desarrollo también.

Cabe destacar que como nuestra REST API y nuestro frontend web estarán corriendo sobre la misma dirección IP y el mismo puerto de escucha. De modo que cuando queramos acceder a una ruta de nuestra API REST desde nuestro proyecto ReactJS basta usar la ruta "/api/controller/value" (por ejemplo).

A partir de ahora, podemos modificar nuestra aplicación ReactJS o ASP.Net Core sin problemas.

Happy coding!!!

ARIEL DAVID (PACO) ESPINOSA PÉREZ
 cl8dep@outlook.com
 @cl8dep

Introducción a XMLA Endpoints en Power BI

En este artículo hablaremos sobre esta nueva funcionalidad de la plataforma Power BI, que nos abre las puertas al mundo de datos a nivel Enterprise. Dicha funcionalidad nos permite separar nuestro modelo de datos de la capa visual y de reportes, con lo que ganamos no solo en prestaciones, sino en gobernabilidad y trabajo en equipo.

Desde hace un tiempo hay empresas que cuentan con modelos de datos, ya sea en SSAS (SQL Server Analysis Services on Premises) o AAS (Azure Analysis Services). Esta funcionalidad les permitirá trabajar con el mismo formato, pero directamente desplegar sus modelos a Power BI.

Conceptos

Comencemos con conceptos básicos de nuestra plataforma Power BI. Por ejemplo, cada vez que comenzamos a desarrollar un nuevo reporte en Power BI Desktop, notamos al abrir nuestro administrador de tareas, que toda sesión de Power BI Desktop maneja un motor tabular de SQL Analysis Services como podemos ver a continuación:

Name	Stat...	21% CPU	46% Memory	1% Disk	1% Network	15% GPU engine
Google Chrome (42)		0.1%	2,125.8 MB	0 MB/s	0 Mbps	0%
Microsoft Edge (9)		0%	119.1 MB	0 MB/s	0 Mbps	0%
Microsoft Outlook		0.2%	190.2 MB	0.1 MB/s	0 Mbps	0.6%
Microsoft PowerPoint		0.2%	208.3 MB	0 MB/s	0 Mbps	0%
Microsoft Teams (7)		4.7%	983.7 MB	0.1 MB/s	0 Mbps	8.3%
Power BI Desktop (10)		0.3%	1,061.6 MB	0 MB/s	0 Mbps	0%
CefSharp.BrowseSubprocess		0%	120.7 MB	0 MB/s	0 Mbps	0%
CefSharp.BrowseSubprocess		0%	6.9 MB	0 MB/s	0 Mbps	0%
CefSharp.BrowseSubprocess		0%	4.5 MB	0 MB/s	0 Mbps	0%
CefSharp.BrowseSubprocess		0%	77.7 MB	0 MB/s	0 Mbps	0%
CefSharp.BrowseSubprocess		0%	74.3 MB	0 MB/s	0 Mbps	0%
CefSharp.BrowseSubprocess		0.1%	189.5 MB	0 MB/s	0 Mbps	0%
CefSharp.BrowseSubprocess		0%	17.7 MB	0 MB/s	0 Mbps	0%
Console Window Host		0%	6.0 MB	0 MB/s	0 Mbps	0%
Microsoft SQL Server Analysis Services		0%	205.1 MB	0 MB/s	0 Mbps	0%
Untitled - Power BI Desktop		0.1%	359.2 MB	0 MB/s	0 Mbps	0%

Imagen 1.- Servicio de SQL Server Analysis Services.

Por otro lado, cada vez que generamos un reporte en Power BI, y dado el ciclo de vida de este, lo publicamos en nuestro Servicio de Power BI, a través de este, y en un área de trabajo (workspace) se genera un nuevo conjunto de datos (o dataset):

NAME	STATUS	OWNER	REFRESH	INCLUDE IN APP
AdventureWorks Data	On	ghowell	On	Yes
AdventureWorks Dataset	On	ghowell	On	Yes
Programad Sample	On	ghowell	On	Yes
Power BI Azure Cognitive Services Image Analysis	On	ghowell	On	No

Imagen 2.- Conjunto de datos generado.

Al trabajar en equipo, una vez el equipo que publica el modelo de datos al servicio de Power BI (lo que se divulga hoy en día como una buena práctica) generar un dataset que puede ser promovido, y luego certificado para que a nivel organizacional se identifique como una fuente confiable de datos. De esta manera, el/los equipos que construyan reportes comienzan su trabajo en Power BI obteniendo datos desde Power BI Dataset (conjunto de datos) conectándose al modelo de datos construido y certificado:

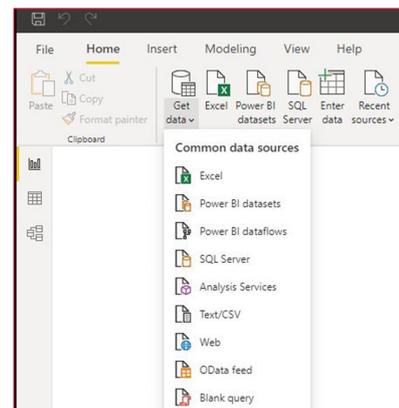


Imagen 3.- Obtención de datos en Power BI Desktop.

Por otro lado, y ahora ya centrándonos en el foco de nuestro artículo el Administrador de Power BI puede habilitar a nivel de nuestro tenant la capacidad de Escritura/Lectura sobre nuestros XMLA Endpoints como podemos ver en la siguiente imagen en el Portal de Administración de Power BI:

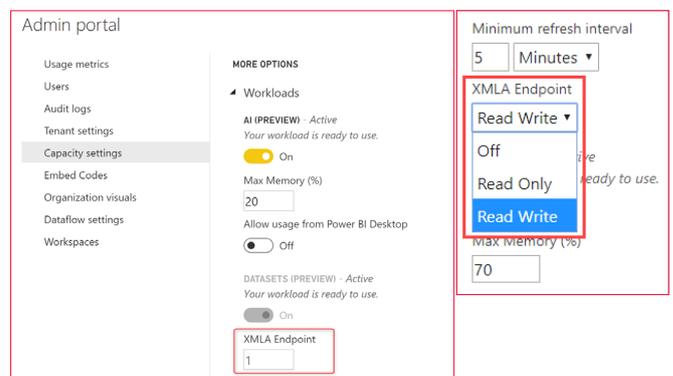


Imagen 4.- Configuraciones en el Portal de Administración de Power BI.

A su vez, cuando habilitamos esta funcionalidad sobre los XMLA Endpoints podremos ver en cada Workspace (o área de trabajo) de Power BI la URL de conexión a nuestro Endpoint que utilizaremos desde varias herramientas:

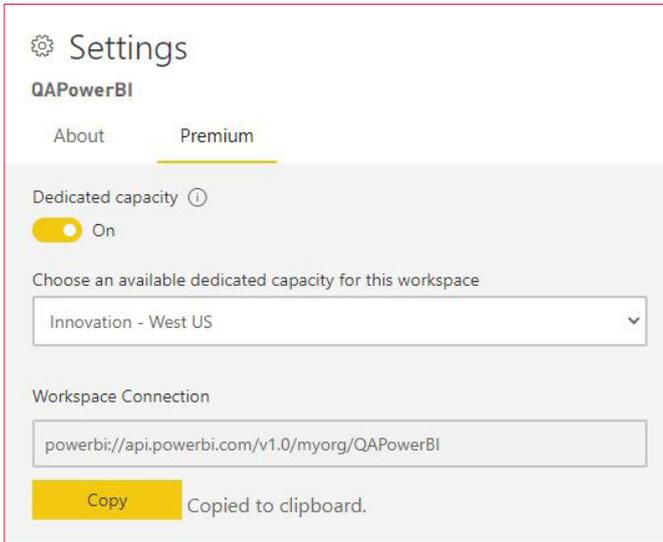


Imagen 5.- URL de conexión a Power BI.

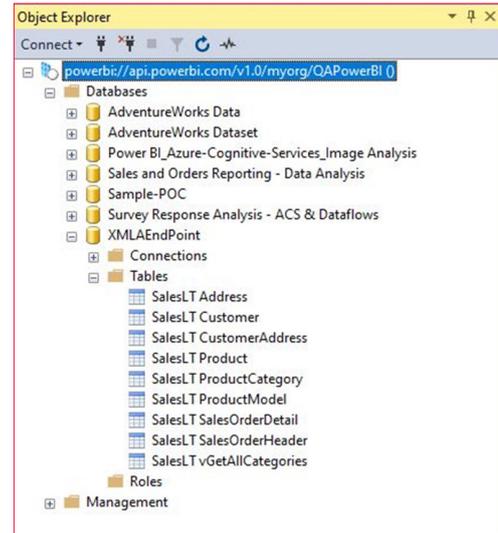


Imagen 7.- Acceso al modelo de datos desde SQL Server Management Studio.

Conexión a XMLA Endpoint desde SSMS (SQL Server Management Studio)

Así como podemos conectarnos a motores de bases de datos en SSMS, también podremos conectarnos a modelos semánticos (Analysis Services). Anteriormente SSMS nos permitía conectarnos a SSAS/AAS, ahora también podemos conectarnos a nuestros modelos de datos (dataset) publicados en Power BI:

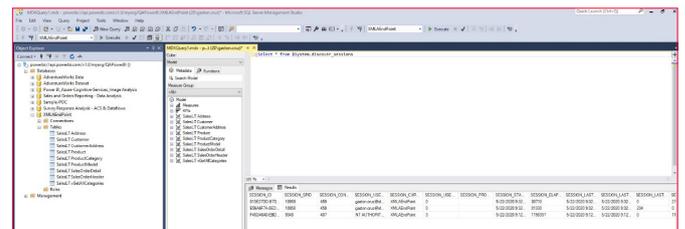
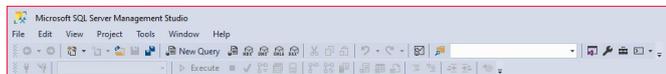


Imagen 8.- Ejecución de consultas en lenguaje XMLA y MDX.

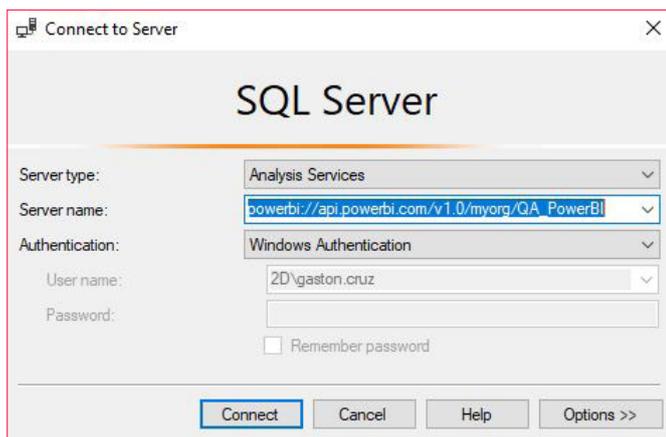


Imagen 6.- Conexión con SQL Server Management Studio.

Implementando Modelos con Visual Studio Data Tools

Esta funcionalidad nos permite desarrollar modelos de datos a través de SSDT (SQL Server Data Tools), lo que nos abre un abanico de opciones a nivel Enterprise muy importante, como por ejemplo vincular nuestra solución a un repositorio de código como Azure DevOps, y generar toda una solución de automatización en materia de despliegues de nuestro modelo a diferentes entornos.

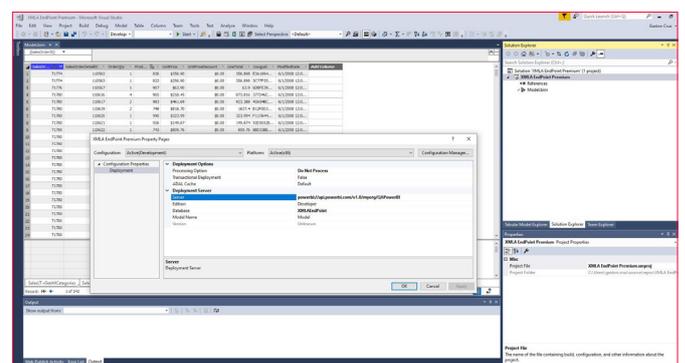


Imagen 9.- Trabajo con Visual Studio Data Tools.

"Power BI toma protagonismo como herramienta de Plataforma de Datos para el manejo de Modelos Semánticos"

Una vez conectados podemos ver todos nuestros modelos datos, y visualizar conexiones, tablas, particiones, roles.

Como podemos ver en la imagen anterior, desde SSDT podemos indicar nuestro endpoint (como servidor) y con

esta opción nuestro despliegue nos generara un conjunto de datos en dicho XMLA endpoint.

Una vez realizado el despliegue, podremos refrescar nuestro dataset, e incluso aprovechar las capacidades de Power BI como agendamiento, como también refrescos incrementales.

Conclusión

Con esta visión Power BI toma protagonismo como herramienta de Plataforma de Datos para el manejo de Modelos Semánticos lo cual nos permite entre otras ventajas: Conectarnos desde SSMS (como vimos anteriormente), tener trazabilidad de consultas a través de SQL Profiler, Desarrollo de Modelos en un equipo integrado de trabajo a través de SSDT, contar con un repositorio de modelos, con versionado, y automatización de procesos con Azure DevOps. Y, por último, los modelos generados podrán ser utilizados

desde múltiples plataformas, y no solo desde Power BI

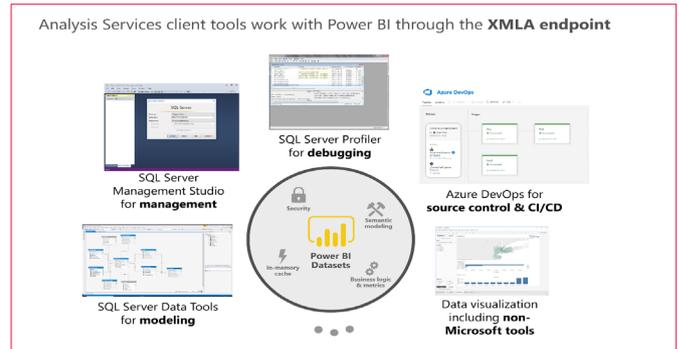


Imagen 10.-Juego de herramientas disponible.

GASTÓN CRUZ

Data Platform MVP

Solution Principal en Slalom USA



Alberto Diaz

Alberto Diaz cuenta con más de 15 años de experiencia en la Industria IT, todos ellos trabajando con tecnologías Microsoft. Actualmente, es Chief Technology Innovation Officer en ENCAMINA, liderando el desarrollo de software con tecnología Microsoft, y miembro del equipo de Dirección.

Desde 2011 ha sido nombrado Microsoft MVP, reconocimiento que ha renovado por séptimo año consecutivo. Se define como un geek, amante de los smartphones y desarrollador. Fundador de TenerifeDev (www.tenerifedev.com), un grupo de usuarios de .NET en Tenerife, y coordinador de SUGES (Grupo de Usuarios de SharePoint de España, www.suges.es)

Email: adiazcan@hotmail.com

Twitter: [@adiazcan](https://twitter.com/adiazcan)



Fabián Imaz

Fabián Imaz, MVP de SharePoint Server trabaja en el mundo del desarrollo de software desde hace más de 10 años, teniendo la suerte de trabajar en distintas arquitecturas y tecnologías Microsoft. Pertenece a la firma Siderys, <http://www.siderys.com> empresa de desarrollo de Software especializada en SharePoint 2007/2010/2013 y en desarrollo de soluciones inteligentes.

Desde los comienzos Fabián ha trabajado en distintas comunidades donde organiza y promueve eventos locales para la difusión de tecnología dentro de los miembros de las mismas. Es director de la carrera SharePoint 2010 y SharePoint 2013 en Microsoft Virtual Academy, <http://www.mslatam.com/latam/technet/mva2/Home.aspx> y cuenta con un sitio en CodePlex con varios desarrollos <http://siderys.codeplex.com>.

Sitio Web: <http://www.siderys.com>

Email: fabiani@siderys.com.uy

Blogs: <http://blog.siderys.com>

Twitter: [@fabianimaz](https://twitter.com/fabianimaz)



Gustavo Velez

Gustavo Velez es Ingeniero Mecánico y Electrónico; trabaja en la arquitectura, diseño e implementación de sistemas de IT basados en tecnologías de Microsoft, especialmente SharePoint, Office 365 y Azure.

Propietario del sitio especializado en información sobre SharePoint en español <http://www.gavd.net>, autor de ocho libros sobre SharePoint y sus tecnologías y numerosos artículos y conferencias sobre el tema.

Sitio Web: <http://www.gavd.net>

Email: gustavo@gavd.net

Blogs: <http://geeks.ms/blogs/gvelez/>



Juan Carlos González Martín

Ingeniero de Telecomunicaciones por la Universidad de Valladolid y Diplomado en Ciencias Empresariales por la Universidad Oberta de Catalunya (UOC). Cuenta con más de 14 años de experiencia en tecnologías y plataformas de Microsoft diversas (SQL Server, Visual Studio, .NET Framework, etc.), aunque su trabajo diario gira en torno a las plataformas SharePoint & Office 365. Juan Carlos es MVP de Office Apps & Services y co-fundador del Grupo de Usuarios de SharePoint de España (SUGES, www.suges.es), del Grupo de Usuarios de Cloud Computing de España (CLOUDES) y de la Comunidad de Office 365. Hasta la fecha, ha publicado 11 libros sobre SharePoint & Office 365, así como varios artículos en castellano y en inglés sobre ambas plataformas.

Email: jcgonzalezmartin1978@hotmail.com

Blogs: <http://geeks.ms/blogs/jcgonzalez> &

<http://jcgonzalezmartin.wordpress.com/>



Santiago Porras

Innovation Team Leader en ENCAMINA, lidera el desarrollo de productos mediante tecnologías Microsoft. Se declara un apasionado de la tecnología, destacando el desarrollo para dispositivos móviles y web, donde ya cuenta con 16 años de experiencia.

Microsoft MVP in Developer Technologies, colabora con las comunidades de desarrolladores desde su blog personal <http://blog.santiagoporras.com> y ocasionalmente en **CompartiMOSS.com**. Además, es uno de los coordinadores de TenerifeDev, grupo de usuarios de .NET en Tenerife (<http://www.tenerifedev.com>)

Sitio Web: <http://www.santiagoporras.com>

Email: santiagoporras@outlook.com

Blogs: <http://blog.santiagoporras.com>

Twitter: [@saintwukong](https://twitter.com/saintwukong)



¿Desea colaborar con CompartiMOSS?



La subsistencia del magazine depende de los aportes en contenido de todos. Por ser una revista dedicada a información sobre tecnologías de Microsoft en español, todo el contenido deberá ser directamente relacionado con Microsoft y escrito en castellano. No hay limitaciones sobre el tipo de artículo o contenido, lo mismo que sobre el tipo de tecnología. Si desea publicar algo, por favor, utilice uno de los siguientes formatos:

- Artículos de fondo: tratan sobre un tema en profundidad. Normalmente entre 2000 y 3000 palabras y alrededor de 4 o 5 figuras. El tema puede ser puramente técnico, tanto de programación como sobre infraestructura, o sobre implementación o utilización.
- Artículos cortos: Artículos cortos: Máximo 1000 palabras y 1 o 2 figuras. Describen rápidamente una aplicación especial de alguna tecnología de Microsoft, o explica algún punto poco conocido o tratado. Experiencias de aplicación en empresas o instituciones puede ser un tipo de artículo ideal en esta categoría.
- Ideas, tips y trucos: Algunos cientos de palabras máximo. Experiencias sobre la utilización de tecnologías de Microsoft, problemas encontrados y como solucionarlos, ideas y trucos de utilización, etc. Los formatos son para darle una idea sobre cómo organizar su información, y son una manera para que los editores le den forma al magazine, pero no son obligatorios. Los artículos deben ser enviados en formato Word (.doc o .docx) con el nombre del autor y del artículo.

Si desea escribir un artículo de fondo o corto, preferiblemente envíe una proposición antes de escribirlo, indicando el tema, aproximada longitud y número de figuras. De esta manera evitaremos temas repetidos y permitirá planear el contenido de una forma efectiva.

Envíe sus proposiciones, artículos, ideas y comentarios a la siguiente dirección:

revista@compartimoss.com

adiazcan@hotmail.com

fabiani@siderys.com.uy

jcgonzalezmartin1978@hotmail.com

gustavo@gavd.net

