

## **Componente .NET para integración con BDU**

# Dirección General de Sistemas de Información y Comunicaciones

## Áreas de Gobierno Tecnológico de SSII y del Dato



Servicio Andaluz de Salud  
Consejería de Sanidad, Presidencia  
y Emergencias

### Contenido

- Dirección General de Sistemas de Información y Comunicaciones
  - Áreas de Gobierno Tecnológico de SSII y del Dato
  - Cumplimiento normativo
  - Histórico de cambios
- SAS.BduApiClient
- Introducción
- Dependencias
- Errores conocidos
- Integración de la librería
- Configuración
  - Prerequisitos
  - Procedimiento de configuración
  - Inyección de dependencias
  - Inyección de dependencias
- Criterios
- Métodos de servicio

### Resumen



- **Versión:** 1.1.0
- **Fecha publicación:** 19 jun 2021
- **Entrada en vigor desde:** 5 ago 2020

blocked URL

- [Uso](#)

## Cumplimiento normativo



Este documento es una guía de desarrollo en continua evolución. Para cualquier duda o sugerencia por favor contante con nosotros.

**Contacto Arquitectura:** [l-arquitectura.stic@juntadeandalucia.es](mailto:l-arquitectura.stic@juntadeandalucia.es)

## Histórico de cambios

| Versión                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 1.4.0 Upgrade | Fecha publicación | 11 may 2021 | Fecha entrada en vigor | 11 may 2021 |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|-------------------|-------------|------------------------|-------------|
| <b>Alcance</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |               |                   |             |                        |             |
| <ul style="list-style-type: none"> <li>Se realiza una reestructuración del proyecto para realizar una división en subcomponentes. De este modo se permite la instalación separada de las implementaciones WebService y Oracle del componente para el acceso a datos de Bdu según sea la necesidad.             <ul style="list-style-type: none"> <li>Los componentes generados han sido los siguientes:                 <ul style="list-style-type: none"> <li>SAS.BduApiClient: Es el proyecto core de la API. Contiene diferentes tipos proporcionados por la API, como BduUser, Address etc, e interfaces para el acceso de datos de Usuarios, mutuas y decretos.</li> <li>SAS.BduApiClient.Exceptions: Manejo de excepciones de negocio de Bdu.</li> <li>SAS.BduApiClient.WebService: Implementación de acceso a los datos de Bdu vía servicio web.</li> <li>SAS.BduApiClient.ManagedDataAccess.Oracle: Implementación vía Réplica de BBDD para acceder en Bdu a datos de mutuas y decretos. En caso de que no se requieran datos de mutuas y decretos no hay que instalar este componente.</li> </ul> </li> </ul> </li> </ul> <p>Se ha cambiado el nombre a la interfaz <b>IBduUserManager</b> y a la clase <b>BduUserManager</b> por <b>IBduUserTransformer</b> y <b>WsBduUserTransformer</b> respectivamente.</p> <p>El tipo <b>EndPointType</b>, en el que se define los tipos de endpoints del servicio web de BDU, se ha llevado al espacio de nombre <b>SAS.BduApiClient.WebService.Api.Entity.EndPointType</b>.</p> |               |                   |             |                        |             |
| Versión                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 1.3.1 Bug fix | Fecha publicación | 09-11-2020  | Fecha entrada en vigor | 09-11-2020  |
| <b>Alcance</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |               |                   |             |                        |             |

- Corrección en la definición de determinadas clases Entity respecto a la definición del esquema de BBDD y envío de la versión de la API en el campo user-agent de la invocación al servicio Web de BDU.
  - Se ha eliminado la definición de esquema de BBDD específico en las siguientes entidades:
    - ComunidadEntity
    - FarmaciaTipoAportacionEntity
    - FinanciacionEntity
    - LocalidadEntity
    - PaisEntity
    - SexoEntity
    - SituacionActualEntity
    - TipoViaEntity
  - En la clase proxy genérico, definida en bduApiClient/WebService/Boundary/WSBduManager.cs, al invocar al servicio web de BDU, se ha añadido la versión actual de la API en el campo User-Agent.

| Versión                                                                                                                                                          | 1.3.0 Upgrade | Fecha publicación | 15-07-2020 | Fecha entrada en vigor | 15-07-2020 |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|-------------------|------------|------------------------|------------|
| Alcance                                                                                                                                                          |               |                   |            |                        |            |
| <ul style="list-style-type: none"> <li>• Añadido consulta para obtener las solicitudes de decreto por réplica de BBDD, en el que se filtra por NUHSA.</li> </ul> |               |                   |            |                        |            |

| Versión                                                                                                                                                                                                                                                                                                                                                                                                                       | 1.2.0 Upgrade | Fecha publicación | 08-06-2020 | Fecha entrada en vigor | 08-06-2020 |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|-------------------|------------|------------------------|------------|
| Alcance                                                                                                                                                                                                                                                                                                                                                                                                                       |               |                   |            |                        |            |
| <ul style="list-style-type: none"> <li>• Añadido posibilidad de definir el endpoint del servicio web de BDU por configuración</li> <li>• Permitir llamadas al servicio web de BDU vía HTTPS</li> <li>• Corregido minúsculas en el espacio de nombre api.boundary. Se ha cambiado a Api.Boundary.</li> <li>• Actualizada versión de MacoApiClient a la v2.4.3, la cual permite llamadas a su servicio web vía HTTPS</li> </ul> |               |                   |            |                        |            |

| Versión                                                                                                                                  | 1.1.0 Upgrade | Fecha publicación | 07-04-2020 | Fecha entrada en vigor | 07-04-2020 |
|------------------------------------------------------------------------------------------------------------------------------------------|---------------|-------------------|------------|------------------------|------------|
| Alcance                                                                                                                                  |               |                   |            |                        |            |
| <ul style="list-style-type: none"> <li>• Añadido dos nuevos servicios para consultar, vía réplica de BBDD, usuarios y mutuas.</li> </ul> |               |                   |            |                        |            |

| Versión                                                                                            | 1.0.3 Daps Update | Fecha publicación | 19-02-2020 | Fecha entrada en vigor | 19-02-2020 |
|----------------------------------------------------------------------------------------------------|-------------------|-------------------|------------|------------------------|------------|
| Alcance                                                                                            |                   |                   |            |                        |            |
| Actualizada la versión del componente MacoApiClient a la v2.4.1, y la versión de oracle a la v19.6 |                   |                   |            |                        |            |

| Versión                                                                                                                                                                                     | 1.0.2 Bug fix | Fecha publicación | 31-07-2019 | Fecha entrada en vigor | 31-07-2019 |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|-------------------|------------|------------------------|------------|
| Alcance                                                                                                                                                                                     |               |                   |            |                        |            |
| <ul style="list-style-type: none"> <li>• Corregido error en el mapeo de los datos de ContactInformation.</li> <li>• Actualizada versión de la librería de MacoApiClient a v2.0.1</li> </ul> |               |                   |            |                        |            |

| Versión                                                                                                                      | 1.0.1 Bug fix | Fecha publicación | 23-04-2019 | Fecha entrada en vigor | 23-04-2019 |
|------------------------------------------------------------------------------------------------------------------------------|---------------|-------------------|------------|------------------------|------------|
| Alcance                                                                                                                      |               |                   |            |                        |            |
| Se actualiza la versión del componente SASUtils.Xml por la v1.0.1, debido a un error en el método obtener texto del nodo XML |               |                   |            |                        |            |

| Versión | 1.0.0 Versión inicial .NET Standard 2.0. | Fecha publicación | 12-03-2019 | Fecha entrada en vigor | 12-03-2019 |
|---------|------------------------------------------|-------------------|------------|------------------------|------------|
| Alcance |                                          |                   |            |                        |            |

Versión inicial de la librería de tipo .NET Standard 2.0

SAS.BduApiClient

## Introducción

En el presente documento se presenta la librería **SAS.BduApiClient**, componente .NET Standar 2.0, para el acceso a datos de BDU.

Esta librería permite el acceso a los datos de BDU utilizando dos implementaciones diferentes: una que utiliza internamente el servicio web de BDU y otra que utiliza réplica de BBDD.

La versión actual de la librería , en su versión acceso a réplica de BBDD, solo dispone de métodos para acceso a la información de mutuas, decretos y los datos principales de usuarios de BDU.

***Se recomienda la instalación de la versión de la librería .NET Standard 2.0, en su versión Servicio Web, compatible con .NET Framework desde la versión 4.6.1 de .NET Framework.***

## Dependencias

| Nombre del componente                    | Versión | Descripción                                                                                                                                             |
|------------------------------------------|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| System.ServiceModel.Duplex               | 4.5.3   | Proporciona clases para consumir y comunicarse con servicios Dúplex                                                                                     |
| System.ServiceModel.Http                 | 4.5.3   | Gestiona solicitudes realizadas a través del protocolo HTTP                                                                                             |
| System.ServiceModel.NetTcp               | 4.5.3   | Genera de forma predeterminada una pila de comunicación en tiempo de ejecución, que utiliza la seguridad de transporte, TCP para la entrega de mensajes |
| System.ServiceModel.Security             | 4.5.3   | Se encarga de temas generales relacionados con la seguridad                                                                                             |
| SAS.Common.Core                          | 1.1.0   | Componente .NET común con clases e interfaces core de aplicaciones                                                                                      |
| SAS.Common.Kernel                        | 1.1.0   | Componente .NET kernel de aplicaciones, como interfaz mapper o diversas utilidades linq                                                                 |
| SAS.MacoApiClient                        | 2.5.0   | Componente core de MacoApiClient                                                                                                                        |
| SAS.MacoApiClient.WebServices            | 2.5.0   | Componente MacoApiClient para el acceso a datos de Maco utilizando los servicios web de Maco                                                            |
| SAS.MacoApiClient.Exceptions             | 2.5.0   | Componente MacoApiClient para el control de excepciones                                                                                                 |
| SASUtils.Parse                           | 1.0.0   | Componente utilidad para la conversión entre tipos                                                                                                      |
| SASUtils.Xml                             | 1.0.1   | Componente .NET común para el tratamiento de XMLs                                                                                                       |
| SAS.Common.OracleProvider                | 1.3.3   | Componente .NET común para la configuración del proveedor de BBDD Oracle                                                                                |
| EntityFramework                          | 6.2.0   | ORM de Microsoft para .NET                                                                                                                              |
| Oracle.ManagedDataAccess                 | 19.6.0  | Proveedor de acceso a datos de Oracle para .NET                                                                                                         |
| Oracle.ManagedDataAccess.EntityFramework | 19.6.0  | ORM de Oracle para .NET                                                                                                                                 |

## Errores conocidos

- Bug al obtener texto de un nodo XML: Corregido desde de la version 1.0.1 

## Integración de la librería

Configurar en Visual Studio un nuevo origen de paquetes Nuget ( clic en “Administrar Paquetes Nuget ...” del proyecto al que se desea añadir la referencia el componente BduApiClient) indicando en el campo Origen la dirección de repositorios de componentes .NET del SAS: <http://net.lib.repository.alm.sas.junta-andalucia.es/>.

### Versión API < 1.4.0

- Instalar el componente SAS.BduApiClient.

### Versión API >= 1.4.0

- Instalar el componente **SAS.BduApiClient.WebService** para el acceso a datos de Bdu utilizando los servicios web de BDU.
- Solo en el caso de que se necesite consultar datos de mutuas y decretos, instalar el componente **SAS.BduApiClient.ManagedDataAccess.Oracle**.

## Configuración

### Prerequisitos

Previo a la configuración de la librería se debe de disponer de los siguientes datos proporcionados por la Oficina Técnica de Interoperabilidad:

- **Código del módulo del sistema en Maco.** Se corresponde con el código en el que está identificado el sistema en Maco.
- **Endpoints del servicio web de Maco**, tanto de PRE como de PRO. En el caso de PRE, es el siguiente:
  - <https://servicios.pre.sas.junta-andalucia.es/usuario/wsBDU.ashx> Servicio web de BDU.

### Procedimiento de configuración

La configuración de la API depende del tipo de implementación de servicios que se vaya a utilizar, según se acceda a los datos mediante el Servicio Web de BDU o mediante réplicas de BBDD. Veamos cada una de ellas:

En ambos ejemplos se utiliza como base un proyecto .NET MVC.

Primero, en el fichero `Startup.cs`, método `ConfigureServices`, se inicializa el contenedor de inyección de dependencia:

#### 1 Inicializar contenedor de inyección de dependencia ContainerDI

```
public void ConfigureServices(IServiceCollection services)
{
    ...
    ContainerDI.SetAsyncScopedLifestyle(); // establecer estilo de vida por defecto del contenedor
    de inyección. Permite procesar tanto operaciones asíncronas como síncronas
    ContainerDI.RegisterServices(ref services);
    ...
}
```

ContainerDI, clase del componente común SAS.Util.Injection, el cual permite el registro por inyección de dependencia de los diferentes servicios y tipos utilizados en el sistema.

- **Configuración de la API para acceder al Servicio Web de BDU**

En el fichero `appsettings.json` se define una sección en la que se establece el código del módulo en MACO y en otra sección, el endpoint del servicio web de BDU.

## 2 Configuración de los parámetros de BDU en fichero de configuración

```
"MacoApiClient": {  
  "MacoModule": "XXX" // el código concreto hay que solicitarlo a la OTI  
},  
"WebServiceLocator": {  
  "ServiceAddressBdu": "https://servicios.pre.sas.junta-andalucia.es/",  
  "EndPointAddressBduSolicitud": "/usuario/wsBDU.ashx"  
}
```

En el fichero Startup.cs, método Configure(IApplicationBuilder app, IHostingEnvironment env), añadir la siguiente configuración de los controladores , viewcomponents de MVC y la llamada al nuevo método InitializeContainer:

## 3 Registrar controladores y ViewComponents

```
public void Configure(IApplicationBuilder app, IHostingEnvironment env)  
{  
  ContainerDI.RegisterMvcControllers(ref app);  
  ContainerDI.RegisterMvcViewComponents(ref app);  
  
  this.InitializeContainer();  
  ...  
}
```

A continuación, en el fichero Startup.cs, crear el método void InitializeContainer() y añadir la siguiente configuración para registrar por inyección de dependencia los diferentes servicios utilizados de la API:



### Inyección de dependencias

En el caso de que no se desee utilizar el inyector de dependencia ContainerDI, la versión 1.4.0 del componente SAS.BduApiClient.WebService permite la configuración de la inyección de dependencia utilizando el inyector por defecto de .NET, del siguiente modo .

```
services.AddBduWSServices(bduApiConfig );
```

Se le pasan como argumentos el objeto configuración de la API. Este método se encuentra en el espacio de nombre SAS.BduApiClient.WebService.IoC.

Configuración del servicio de búsqueda de BDU que utiliza internamente el servicio web de BDU

## 4 Registrar el servicio de búsqueda de BDU

```
private void InitializeContainer()  
{  
  #region Registrar BDU  
    ContainerDI.Register<IBduSearch<MacoUser>, BduSearch>(ContainerDI.  
LifeStyleContainer.Transient);  
  }
```

Configuración de la API de BDU, indicando url principal de BDU y código de la aplicación en MACO

## 5 Configuración de la API de BDU

```
var bduApiConfig = new BduApiConfig(Configuration.GetSection("WebServiceLocator:ServiceAddressBdu").Value, Configuration.GetSection("MacoApiClient:MacoModule").Value){};
```

Configuración el endpoint del Servicio web de BDU

## 6 Configuración del endpoint del servicio de BDU

```
bduApiConfig.EndPoints = new Dictionary<BduApiClient.WebService.Api.Entity.EndPointType, string>
{
    { SAS.BduApiClient.WebService.Api.Entity.EndPointType.Solicitud, Configuration.GetSection("WebServiceLocator:EndPointAddressBduSolicitud").Value }
};
```



### Breaking changes

En las versiones anteriores a 1.4.0, el enumerado `EndPointType` tenía el espacio de nombre `SAS.BduApiClient.Api.Entity` .

Registro del tipo `BduApiConfig` que implementa la interfaz de configuración de BDU, `IBduApiConfig`

## 7 Registro del tipo `BduApiConfig` que implementa la interfaz `IBduApiConfig`

```
ContainerDI.Register<IBduApiConfig>(() => bduApiConfig, ContainerDI.LifeStyleContainer.Transient);
```

Registro del tipo `WSBduUserTransformer`, que implementa la interfaz `IBduUserTransformer`. Realiza la transformación de los mensajes XML recibidos del servicio web de BDU a objetos.

## 8 Registrar el servicio de transformación de datos de BDU

```
ContainerDI.Register<IBduUserTransformer , WSBduUserTransformer >(ContainerDI.LifeStyleContainer.Transient);
```



#### Breaking changes

En la versión 1.4.0, se ha cambiado el nombre a la interfaz `IBduUserManager` y a la clase `BduUserManager` por `IBduUserTransformer` y `WSBduUserTransformer` respectivamente.

Registro del tipo `BduExceptionHandler`, que implementa la interfaz `IBduExceptionHandler`. Se encarga de gestionar las excepciones y errores informados del servicio web de BDU.

#### 9 Registrar el tipo `BduExceptionHandler`

```
ContainerDI.Register<IBduExceptionHandler, BduExceptionHandler>(ContainerDI.LifeStyleContainer.Transient);
```

Registro del tipo `WSBduManager`, que implementa la interfaz `IWSBduManager`. Es la clase wrapper encargada de llamar a los servicios web de BDU.

#### 10 Registrar el servicio `WSBduManager`

```
ContainerDI.Register<IWSBduManager, WSBduManager>(ContainerDI.LifeStyleContainer.Transient);  
}
```

- Configuración de la API versión Réplica de BBDD

Registro del servicio `IConfigContextApiConnection`, configuración del contexto de BBDD con el tipo concreto `ConfigContextApiBduConnection`. Recibe como parámetros la cadena de conexión de la réplica, en este caso del fichero de configuración `app.config`, tipo de esquema BDU y nombre del esquema de la BBDD.

#### 11 Registrar el contexto de BBDD de Bdu

```
private void InitializeContainer()
{
...
    #region Registrar BDU
    // Conexion a la réplica de BDU

    var configContextApiBduConnection = new ConfigContextApiBduConnection(new
ManagedConfigConnectionBase<SchemaType>[]
    {
        new ManagedConfigConnectionBase<SchemaType>(ConfigurationManager.ConnectionStrings
["BDUConnection"].ConnectionString, SchemaType.Bdu, "NOMBRE DEL ESQUEMA")
    });
    ContainerDI.Register<IConfigContextApiConnection<SchemaType>>(() =>
configContextApiBduConnection , ContainerDI.LifeStyleContainer.Transient);
```

Registro del servicio BduUserService, que implementa la interfaz IBduSearch. Permite el acceso a datos de usuario vía réplica de BBDD.

#### 12 Registrar el servicio BduUserService

```
ContainerDI.Register<IBduSearch<object>, BduUserService>(ContainerDI.LifeStyleContainer.
Transient);
```

Registro del servicio MutualInsuranceService, que implementa la interfaz IMutualInsuranceService, para el acceso a datos de mutuas vía réplica de BBDD.

#### 13 Registrar el servicio MutualInsuranceService

```
ContainerDI.Register<IMutualInsuranceService<object>, MutualInsuranceService>(ContainerDI.
LifeStyleContainer.Transient);
```

Registro del servicio BduDecreeRequestService, que implementa la interfaz IBduDecreeRequestService, para el acceso a datos de decretos vía réplica de BBDD.

#### 14 Registrar el servicio BduDecreeRequestService

```
ContainerDI.Register<IBduDecreeRequestService<object>, BduDecreeRequestService>(ContainerDI.
LifeStyleContainer.Transient);
```

Registro del tipo UserRepository que implementa la interfaz IUserRepository. Repositorio encargado de acceder a datos de usuario vía réplica de BBDD.

#### 15 Registrar el repositorio UserRepository

```
ContainerDI.Register<IUserRepository, UserRepository>(ContainerDI.LifeStyleContainer.  
Transient);
```

Registro del tipo MutualidadRepository que implementa la interfaz IMutualidadRepository. Repositorio encargado de acceder a datos de mutualidad vía réplica de BBDD.

#### 16 Registrar el repositorio MutualidadRepository

```
ContainerDI.Register<IMutualidadRepository, MutualidadRepository>(ContainerDI.  
LifeStyleContainer.Transient);  
  
#endregion  
...  
}
```



### Inyección de dependencias

En el caso de que no se desee utilizar el inyector de dependencia ContainerDI, la versión 1.4.0 del componente SAS.BduApiClient.ManagedDataAccess.Oracle permite la configuración de la inyección de dependencia utilizando el inyector por defecto de .NET, del siguiente modo.

```
services.AddBduOracleServices(configContextApiBduConnection );
```

Se le pasa como argumento la configuración del contexto de BDU. Este método se encuentra en el espacio de nombre SAS.BduApiClient.ManagedDataAccess.Oracle.IoC.

A continuación se presenta el código completo del ejemplo de configuración :

#### 17 Ejemplo de configuración de la librería utilizando el inyector de dependencia ContainerDI

```
public class Startup  
{  
    public Startup(IConfiguration configuration)  
    {  
        Configuration = configuration;  
    }  
  
    public IConfiguration Configuration { get; }  
  
    // This method gets called by the runtime. Use this method to add services to the container.  
    public void ConfigureServices(IServiceCollection services)  
    {  
        ...  
    }  
}
```

```

        ContainerDI.SetAsyncScopedLifestyle();
        ContainerDI.RegisterServices(ref services);
    }

    // This method gets called by the runtime. Use this method to configure the HTTP request
    pipeline.
    public void Configure(IApplicationBuilder app, IHostingEnvironment env)
    {
        ContainerDI.RegisterMvcControllers(ref app);
        ContainerDI.RegisterMvcViewComponents(ref app);

        app.UseDeveloperExceptionPage();

        this.InitializeContainer();

        ...
    }

    /// <summary>
    /// Inicializador de contenedor utilizando el inyector de dependencia ContainerDI.
    /// </summary>
    private void InitializeContainer()
    {
        ...
        #region registrar BDU
        ContainerDI.Register<IUserService, UserService>(ContainerDI.LifeStyleContainer.
Transient);
        ContainerDI.Register<IBduUseCase, BduUseCase>(ContainerDI.LifeStyleContainer.
Transient);
        ContainerDI.Register<IBduSearch<MacoUser>, BduSearch>(ContainerDI.
LifeStyleContainer.Transient);

        var bduApiConfig = new BduApiConfig(Configuration.GetSection("WebServiceLocator:
ServiceAddressBdu").Value, Configuration.GetSection("MacoApiClient:MacoModule").Value)
        {
        };
        bduApiConfig.EndPoints = new Dictionary<BduApiClient.WebService.Api.Entity.
EndPointType, string>
        {
            { SAS.BduApiClient.WebService.Api.Entity.EndPointType.Solicitud,
Configuration.GetSection("WebServiceLocator:EndPointAddressBduSolicitud").Value }
        };

        ContainerDI.Register<IBduApiConfig>(() => bduApiConfig, ContainerDI.
LifeStyleContainer.Transient);
        ContainerDI.Register<IBduUserTransformer, WSBduUserTransformer>(ContainerDI.
LifeStyleContainer.Transient);
        ContainerDI.Register<IBduExceptionHandler, BduExceptionHandler>(ContainerDI.
LifeStyleContainer.Transient);
        ContainerDI.Register<IBduService, BduService>(ContainerDI.LifeStyleContainer.
Transient);

        ContainerDI.Register<IWsBduManager, WsBduManager>(ContainerDI.LifeStyleContainer.
Transient);

        ContainerDI.Register<IBduFactory>(() => new BduFactory(bduApiConfig),
ContainerDI.LifeStyleContainer.Transient);

        //descomentar en caso de consultar datos de mutuas vía réplica de BBDD
        //var configContextApiBduConnection = new ConfigContextApiBduConnection(new
ManagedConfigConnectionBase<SchemaType>[]
        //{
            //new ManagedConfigConnectionBase<SchemaType>(ConfigurationManager.
ConnectionStrings["BDUConnection"].ConnectionString, SchemaType.Bdu, "NOMBRE DEL
ESQUEMA")
        //});
        //ContainerDI.Register<IConfigContextApiConnection<SchemaType>>(() =>
configContextApiBduConnection , ContainerDI.LifeStyleContainer.Transient);
        //ContainerDI.Register<IBduSearch<object>, BduUserService>(ContainerDI.

```

```

LifeStyleContainer.Transient);
        //ContainerDI.Register<IMutualInsuranceService<object>, MutualInsuranceService>
(ContainerDI.LifeStyleContainer.Transient);

        //ContainerDI.Register<IBduDecreeRequestService<object>,
BduDecreeRequestService>(ContainerDI.LifeStyleContainer.Transient);
        //ContainerDI.Register<IUserRepository, UserRepository>(ContainerDI.
LifeStyleContainer.Transient);
        //ContainerDI.Register<IMutualidadRepository, MutualidadRepository>(ContainerDI.
LifeStyleContainer.Transient);

        #endregion registrar BDU
    }
}

```

#### 18 Ejemplo de configuración de la librería utilizando el inyector de dependencia por defecto de .NET

```

    public class Startup
    {
        public Startup(IConfiguration configuration)
        {
            Configuration = configuration;
        }

        public IConfiguration Configuration { get; }

        // This method gets called by the runtime. Use this method to add services to the container.
        public void ConfigureServices(IServiceCollection services)
        {
            ...
            #region configuración BDU
            var bduApiConfig = new BduApiConfig(Configuration.GetSection("WebServiceLocator:
ServiceAddressBdu").Value, Configuration.GetSection("MacoApiClient:MacoModule").Value)
            {
            };
            bduApiConfig.EndPoints = new Dictionary<BduApiClient.WebService.Api.Entity.
EndPointType, string>
            {
                { SAS.BduApiClient.WebService.Api.Entity.EndPointType.Solicitud,
Configuration.GetSection("WebServiceLocator:EndPointAddressBduSolicitud").Value }
            };
            services.AddBduWSServices(bduApiConfig);

            //descomentar en caso de consultar datos de mutuas vía réplica de BBDD
            //var configContextApiBduConnection = new ConfigContextApiBduConnection(new
ManagedConfigConnectionBase<SchemaType>[]
            //{
                //new ManagedConfigConnectionBase<SchemaType>(ConfigurationManager.ConnectionStrings
["BDUConnection"].ConnectionString, SchemaType.Bdu, "NOMBRE DEL ESQUEMA")
            //});
            //services.AddBduOracleServices( configContextApiBduConnection );

            #endregion
        }
    }

```

## Criterios

El objeto **QuerysBduUserSearchDto**, es la representación de los criterios de búsqueda que son soportados por la API client. Dicho objeto permite los siguientes valores:

- Flags, estos son los modificadores soportados:
  - DetailedInfo: Booleano indicando si se desea la información detallada (true) o resumida (false). Valor por defecto: False
  - FarmacialInfo: Booleano indicando si se desea la información detallada sobre farmacia. Valor por defecto: False
  - FullAddress: Booleano indicando si se desea la información detallada del domicilio del paciente. Valor por defecto: False.
- Identificadores de usuario único
  - NUHSA: Alfanumérico.
  - CodSns: Alfanumérico.
  - SocialSecurityNumber: Alfanumérico.
  - AffiliationNumber: Alfanumérico
  - Nuhsas: Lista de códigos NUHSA
  - DocumentIdType + DocumentType + DocumentId: permite identificar un paciente por el identificador definido.
- Filtros de búsqueda
  - BirthDateYear: Alfanumérico.
  - Name: Nombre del paciente
  - Surname: Primer apellido del paciente
  - SecondSurname: Segundo apellido del paciente

Antes de realizar la búsqueda de usuarios vía réplica de BBDD, se valida que el objeto criterio de búsqueda **QuerysBduUserSearchDto** cumpla las siguientes condiciones:

- El objeto **QuerysBduUserSearchDto** no puede ser nulo
- No pueden ser nulo los siguientes atributos: **SocialSecurityNumber**, **AffiliationNumber**, **NUHSA**, **CodSns**, **DocumentId**
- Al menos se deben de especificar dos campos de entre: **Name**, **Surname**, **SecondSurname**

## Métodos de servicio

Los métodos de servicio que la API proporciona son los siguientes:

- Búsqueda de usuarios (consula detallada/corta)

La búsqueda de usuarios se realizará mediante los métodos "findUsers" que soportan el criterio descrito anteriormente

|                                                                                   |                                                                                                                                                                                                                          |
|-----------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | <p><b><u>Breaking changes</u></b></p> <p>La versión actual del componente <code>SAS.BduApiClient.ManagedDataAccess</code>.<br/>Oracle Actualmente, versión réplica de la API, sólo se permite la consulta detallada.</p> |
|-----------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

- Búsqueda información de mutuas (sólo modo réplica de BBDD)
  - Listado de mutualidades
  - Listado de mutualidades que cumplen con un criterio
- Búsqueda información de decretos (sólo modo réplica de BBDD)
  - Búsqueda de información de solicitud de decreto que cumple un criterio

## Uso

A continuación se describe el uso de la librería.

### Funcionalidades utilizando Servicio Web de BDU

- Búsqueda de usuarios

Una vez realizada la configuración de la API, se siguen los siguientes pasos para realizar una búsqueda de usuarios utilizando el servicio Web de BDU.

En la clase donde se va a añadir esta funcional, BduService en el ejemplo, añadir al constructor un objeto de tipo IBduSearch: IBduSearch<MacoUser> bduSearch.

#### 19 Constructor de la clase que llamará a los métodos de la API BDU

```
...
public class BduService: IBduService
{
    private IBduSearch<MacoUser> _bduSearch;
    public BduService(IBduSearch<MacoUser> bduSearch)
    {
        _bduSearch = bduSearch;
    }
    ...
}
```

Añadir un método, FindUsers en el ejemplo, con los siguientes parámetros: IGenericAuthContext<MacoUser> petitionerUser y QuerysBduUserSearchDto querysBduUserSearchDto. El primero hace referencia al usuario Maco petionario de la llamada y el segundo los criterial de la query que se va a realizar.

Llamar al método FindUsers del servicio de BDU, \_bduSearch en el ejemplo, pasando como parámetro el usuario petionario y la query, petitionerUser y queryBduUserSearchDto en el ejemplo.

Por último, se hace el mapping entre el tipo usuario BDU devuelto por la API y el propio del sistema que realiza la llamada.

## 20 Método que busca usuarios en BBDD

```
public List<BduUser> FindUsers(IGenericAuthContext<MacoUser> petitionerUser,
QueryysBduUserSearchDto queryysBduUserSearchDto)
{
    List<BduUser> resultFindUsers = new List<BduUser>();
    var result = _bduSearch.FindUsers(petitionerUser, queryysBduUserSearchDto);
    foreach (SAS.BduApiClient.Api.Entity.BduUser bduUser in result)
    {
        var bduUserLocal = new BduUser();
        bduUserLocal.Name = bduUser.Name;
        bduUserLocal.FirstSurname = bduUser.FirstSurname;
        bduUserLocal.SecondSurname = bduUser.SecondSurname;
        //en un mapping hacer resto de campos
        resultFindUsers.Add(bduUserLocal);
    }
    return resultFindUsers;
}
```

El código completo del ejemplo que consulta datos de usuarios vía Réplica de BBDD se muestra a continuación:

## 21 Ejemplo completo de búsqueda de usuarios en BBDD

```
using SAS.MacoApiClient.User.Entity;
using SAS.Common.Core.Boundary.Contracts;
using SAS.BduApiClient.Api.Boundary;
using SAS.BduApiClient.Api.Entity.dto;

public class BduService: IBduService
{
    private IBduSearch<MacoUser> _bduSearch;
    public BduService(IBduSearch<MacoUser> bduSearch)
    {
        _bduSearch = bduSearch;
    }

    public List<BduUser> FindUsers(IGenericAuthContext<MacoUser> petitionerUser,
QueryysBduUserSearchDto queryysBduUserSearchDto)
    {
        List<BduUser> resultFindUsers = new List<BduUser>();
        var result = _bduSearch.FindUsers(petitionerUser, queryysBduUserSearchDto);
        foreach (SAS.BduApiClient.Api.Entity.BduUser bduUser in result)
        {
            var bduUserLocal = new BduUser();
            bduUserLocal.Name = bduUser.Name;
            bduUserLocal.FirstSurname = bduUser.FirstSurname;
            bduUserLocal.SecondSurname = bduUser.SecondSurname;
            //en un mapping hacer resto de campos
            resultFindUsers.Add(bduUserLocal);
        }
        return resultFindUsers;
    }
}
```

Crear la interfaz IBduService que implementa la clase BduService:

## 22 Definición de la interfaz IBduService

```
public interface IBduService
{
    List<BduUser> FindUsers(IGenericAuthContext<MacoUser> petitionerUser, QuerysBduUserSearchDto
querysBduUserSearchDto);
}
```

La clase que se haya creado para añadir la funcionalidad de búsqueda de usuarios se debe registrar en la clase Startup.cs, método InitializeContainer(), al igual que el resto de servicio y tipos. Continuando con el ejemplo:

## 23 Registrar la clase servicio que implementa la interfaz IBduService con el inyector de dependencia ContainerDI

```
ContainerDI.Register<IBduService, BduService>(ContainerDI.LifeStyleContainer.Transient);
```

o si se declara con el inyector de dependencia por defecto de .NET:

## 24 Registrar la clase servicio que implementa la interfaz IBduService con el inyector de dependencia por defecto de .NET

```
services.AddTransient<IBduService, BduService>();
```

## Funcionalidades utilizando la réplica de BDU

- **Búsqueda de usuarios**

La búsqueda de usuarios se realiza del mismo modo que se explica en el punto anterior. Únicamente hay que tener en cuenta que el servicio que implementa la interfaz IBduSearch es, en este caso, SAS.BduApiClient.ManagedDataAccess.Oracle.Boundary.BduUserService (ver **Configuración de la API versión Réplica de BBDD**) y que en modo réplica, y en la versión actual de la librería, sólo se puede especificar la consulta detallada.

- **Información de Mutuas**

Para consultar información de mutuas, se realiza los siguientes pasos:

En la clase donde vamos a llamar a los diferentes métodos de la API de BDU, BduService en el ejemplo, definimos un objeto interfaz de tipo servicio de mutuas IMutualInsuranceService. Será el objeto utilizado para acceder a los datos de mutuas.

En el método constructor de la clase donde vamos a utilizar la funcionalidad de mutuas, se añade un objeto de tipo IMutualInsuranceService: IMutualInsuranceService<object> servicioMutua;

## 25 Constructor de la clase que llamará a los métodos de la API BDU

```
public class BduService : IBduService
{
    /// <summary>
    /// Interfaz de servicio al componente de bdu para la mutuas
    /// </summary>
    private readonly IMutualInsuranceService<object> servicioMutua;

    public BduService( IMutualInsuranceService<object> servMutua)
    {
        this.servicioMutua = servMutua;
    }
}
```

- **Listado de mutuas.**

A continuación se llama al método `GetMutualInsurance` del servicio, pasando como parámetro peticionario un `null`: `servicioMutua.GetMutualInsurance(null)`;, tal y como se observa en el método `GetMutuas()` del ejemplo:

## 26 Método que obtiene listado mutuas de BBDD

```
/// <summary>
/// Método que obtiene todas las mutuas.
/// </summary>
/// <returns>Listado de mutuas aseguradoras.</returns>
public List<MutuaAseguradora> GetMutuas()
{
    var mut = this.servicioMutua.GetMutualInsurance(null);
    List<MutuaAseguradora> domMut = new List<MutuaAseguradora>();

    if (mut != null)
    {
        foreach (var item in mut)
        {
            domMut.Add(new MutuaAseguradora(item.Id,item.Description));
        }
    }

    return domMut;
}
```



### Breaking changes

El parámetro peticionario es `null` debido a que en la réplica no se registra peticionario de la consulta. En un futuro servicio web de mutuas sí podría ser necesario el parámetro peticionario definido en la interfaz para su registro.

- **Búsqueda de mutuas que cumplen una determinada condición**

En este caso se llamaría al método FindMutualInsurances, que recibe como parámetros el peticionario y el criterio de búsqueda de mutuas, tal y como se observa en el método GetMutuaById del ejemplo::

#### 27 Método que obtiene mutua por id

```
/// <summary>
/// Método que obtiene una mutua por su codigo de identificacion.
/// </summary>
/// <returns>Codigo y descripcion de la mutua aseguradora.</returns>
public MutuaAseguradora GetMutuaById(decimal? codigo)
{
    if (codigo != null)
    {
        var item= this.servicioMutua.FindMutualInsurances(null, new QuerysMutualInsuranceDto {
Id = codigo ?? 0 });
        if (item!= null) return new MutuaAseguradora(item.Id,item.Description);
        else
            return null;
    }
    else
        return null;
}
```

- **Información de decretos**

Para consultar información de decretos, realizamos los siguientes pasos:

En el constructor de la clase donde vamos a utilizar la funcionalidad de decretos, se añade un objeto de tipo IBduDecreeRequestService: IBduDecreeRequestService<object> servicioDecretos;

#### 28 Constructor de la clase que llamará a los métodos de la API BDU

```
public class BduService : IBduService {
    /// <summary>
    /// Interfaz de servicio al componente de bdu para las mutuas.
    /// </summary>
    private readonly IBduDecreeRequestService<object> servicioDecretos;

    public BduService( IBduDecreeRequestService<object> servicioDecretos)
    {
        this.servicioMutua = servMutua;
    }

    ...
}
```

A continuación se llama al método public BduDecreeRequest FindBduElementsDecreeRequest (IGenericAuthContext<object> petitionerBdu, QuerysBduUserSearchDto criteria), que recibe como parámetros el peticionario y el criterio de búsqueda.

En el atributo peticionario se le pasaría un null y en criteria el criterio de búsqueda QuerysBduUserSearchDto, con el NUHSA relleno. Devuelve un objeto petición de decreto con los siguientes valores de la tabla BDU por NUHSA:

- SITUACION\_ACTUAL: Situacion del paciente en BDU en el momento de la inscripcion en el Registro (0=Vigente, 1=Pasivo)
- CLAVE\_MEDICA: Clave medica del paciente en BDU en el momento de la inscripcion en el Registro
- VALIDADO: Indica si los datos del paciente en BDU estaban validados en el momento de la inscripcion en el Registro (0=Si, 1=No)
- ESPECIALIDAD\_CM: Especialidad de la clave medica del paciente en BDU en el momento de la inscripcion (G=General, P=Pediatrica)

#### 29 Método búsqueda de petición de decretos

```
List<BduDecreeRequest> FindDecreeRequest( QuerysBduUserSearchDto querysBduUserSearchDto)
{
    var listaBduDecreeRequest = this.servicioDecretos.FindBduElementsDecreeRequest(null,
querysBduUserSearchDto);
    List<PeticionDecreto> listaPeticionDecreto = new List<PeticionDecreto>();

    if (listaBduDecreeRequest != null)
    {
        foreach (var item in listaBduDecreeRequest)
        {
            var peticionDecreto = new PeticionDecreto()
            {
                Nuhsa = item.Nusa,
                ClaveMedica= item.Clave_Medica,
                EspecialidadCm = item.Especialidad_Cm,
                SituacionActual = item.Situacion_Actual,
                Validado = item.Validado
            };

            listaPeticionDecreto.Add( peticionDecreto );
        }
    }

    return listaPeticionDecreto;
}
```

El código completo del ejemplo de consulta de datos de mutuas y decretos se muestra a continuación:

#### 30 Ejemplo de búsqueda de mutuas y decretos

```
public class BduService : IBduService {

    /// <summary>
    /// Interfaz de servicio al componente de bdu para la mutuas
    /// </summary>
    private readonly IMutualInsuranceService<object> servicioMutua;

    /// <summary>
```

```

/// Interfaz de servicio al componente de bdu para las decretos.
/// </summary>
private readonly IBduDecreeRequestService<object> servicioDecretos;

public BduService( IMutualInsuranceService<object> servMutua, IBduDecreeRequestService<object>
servicioDecretos)
{
    this.servicioMutua = servMutua;
    this.servicioDecretos = servicioDecretos;
}

/// <summary> /// Método que obtiene todas las mutuas.
/// </summary> /// <returns>Listado de mutuas aseguradoras.</returns>
public List<MutuaAseguradora> GetMutuas() {
    var mut = this.servicioMutua.GetMutualInsurance(null);
    List<MutuaAseguradora> domMut = new List<MutuaAseguradora>();

    if (mut != null) {
        foreach (var item in mut) {
            domMut.Add(new MutuaAseguradora(item.Id,item.Description));
        }
    }
    return domMut;
}

/// <summary>
/// Método que obtiene una mutua por su codigo de identificacion.
/// </summary>
/// <returns>Codigo y descripcion de la mutua aseguradora.</returns>
public MutuaAseguradora GetMutuaById(decimal? codigo)
{
    if (codigo != null)
    {
        var item= this.servicioMutua.FindMutualInsurances(null, new QuerysMutualInsuranceDto {
Id = codigo ?? 0 });
        if (item!= null) return new MutuaAseguradora(item.Id,item.Description);
        else
            return null;
    }
    else
        return null;
}

List<BduDecreeRequest> FindDecreeRequest( QuerysBduUserSearchDto querysBduUserSearchDto)
{
    var listaBduDecreeRequest = this.servicioDecretos.FindBduElementsDecreeRequest(null,
querysBduUserSearchDto);
    List<PeticionDecreto> listaPeticionDecreto = new List<PeticionDecreto>();

    if (listaBduDecreeRequest != null)
    {
        foreach (var item in listaBduDecreeRequest)
        {
            var peticionDecreto = new PeticionDecreto()
            {
                Nuhsa = item.Nusa,
                ClaveMedica= item.Clave_Medica,
                EspecialidadCm = item.Especialidad_Cm,
                SituacionActual = item.Situacion_Actual,
                Validado = item.Validado
            };

            listaPeticionDecreto.Add( peticionDecreto );
        }
    }
}

```

```
}    return listaPeticionDecreto;
```