

Componente .NET para el acceso a datos de Estructura

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
- Introducción
- Dependencias
- Errores conocidos
- Historial de cambios
- Integración de la librería
 - Configuración
 - Versión <= 1.1.2
 - Versión 1.2
 - Versión >= 1.3
- Uso

Resumen



- **Versión:** 1.0.0
- **Fecha publicación:** 5 ago 2020
- **Entrada en vigor desde:** 5 ago 2020

blocked URL

Cumplimiento normativo



Las normas expuestas son de obligado cumplimiento. La STIC podrá estudiar los casos excepcionales los cuales serán gestionados a través de los responsables del proyecto correspondiente y autorizados por el Área de Gobernanza de la STIC. Asimismo cualquier aspecto no recogido en estas normas deberá regirse en primera instancia por las guías técnicas correspondientes al esquema nacional de seguridad y esquema nacional de interoperabilidad según correspondencia y en su defecto a los marcos normativos y de desarrollo software establecidos por la Junta de Andalucía, debiendo ser puesto de manifiesto ante la STIC.

La STIC se reserva el derecho a la modificación de la norma sin previo aviso, tras lo cual, notificará del cambio a los actores implicados para su adopción inmediata según la planificación de cada proyecto.

En el caso de que algún actor considere conveniente y/o necesario el incumplimiento de alguna de las normas y /o recomendaciones, deberá aportar previamente la correspondiente justificación fehaciente documentada de la solución alternativa propuesta, así como toda aquella documentación que le sea requerida por la STIC para proceder a su validación técnica.

Contacto Arquitectura: l-arquitectura.stic@juntadeandalucia.es

Histórico de cambios

Los cambios en la normativa vendrán acompañados de un registro de las modificaciones. De este modo se podrá realizar un seguimiento y consultar su evolución. Ordenándose de mas recientes a menos recientes, prestando especial cuidado a las cabezas de la tablas dónde se indican las fechas de entrada en vigor y versión.

Versión	v02r00	Fecha publicación	26 de octubre de 2018	Fecha entrada en vigor	1 de enero de 2019
Alcance					
- Se actualizan todas las referencias al repositorio de código SVN incluyendo GIT como nuevo repositorio. - Dentro del entorno tecnológico se incluye un enlace para la descarga de una instancia de Sonarqube 6.7.2 parametrizada con los datos de la STIC <u>Características del fichero de configuración</u>: se ha incluido una tabla con los códigos a incluir en la propiedad groupId. Esta tabla contiene una codificación de las diferentes suite de aplicaciones existentes en la CMS - Se ha incluido cual es objeto de las <u>PLs de "Correctivo de datos" y "Carga de datos"</u> y cual es el contenido de los scripts que se adjuntan <u>Verificación de calidad del código fuente</u>: se indican cuales son la nuevas métricas Sonarqube a revisar, umbrales mínimo y de confianza y los criterios para la revisión de PLs en FARO <u>Calidad de test unitarios</u>: para aquellos proyectos JAVA se incluye el plugin PITEST como parte de la ejecución en JENKINS con el objeto de medir la calidad de los test unitarios realizados. <u>Normativa CNO</u>: será de obligatorio cumplimiento todas las reglas descritas en la herramienta CNO <u>Ficha de Entrega en PLs</u>: Se elimina la obligatoriedad de adjuntar la ficha de entrega en las PLs de FARO. Esta información es mantenida por el proveedor de desarrollo en GIT mediante el fichero readme.md					
Versión	v03r00	Fecha publicación		Fecha entrada en vigor	
Alcance					

- Verificación de calidad del código fuente: se indican cuales son la nuevas métricas Sonarqube a revisar, umbrales mínimo y de confianza y los criterios para la revisión de PLs en FARO

Introducción

En el presente documento se presenta la librería **SAS.EstructuraApiClient**, componente .NET Standar 2.0, para el acceso a datos de Estructura. Esta librería permite el acceso a los datos de Estructura utilizando por ahora réplica de BBDD.

Dependencias

Nombre del componente	Versión	Descripción
SAS.MacoApiClient	2.4.3	Componente .NET Standard para el acceso a datos de Maco.
EntityFramework	6.2.0	ORM de Microsoft .NET
SAS.Common.OracleProvider	1.3.3	Componente .NET común para la configuración del proveedor de BBDD Oracle
SAS.Common.Kernel	1.1.0	Componente .NET común kernel de aplicaciones: interfaz de mapper, extensión de linq, etc
SAS.Common.Core	1.1.0	Componente .NET común con clases e interfaces core de aplicaciones
SAS.Util.Net.Http	1.2.1	Componente .NET configuración de clientes Http para comunicación con servicios REST
HL7.Fhir.R4	1.4.0	Componente de la versión R4 de la especificación Fhir

Errores conocidos

Historial de cambios

Histórico de cambios realizados en la librería

- 1.0.0 - Versión inicial.
- 1.0.1 - Versión MPA.
- 1.1.0 - Versión Citación.
- 1.1.1 - Se actualiza versión MacoApiClient a la 2.0.1, por un error en éste
- 1.1.2 - Se añade métodos para nueva solicitud de citación
- 1.2.0 - Cambios en la declaración de configuración de conexión
- 1.3.0 - Reorganización de estructura de proyectos separando: Core, API y ManagedDataAccess.Oracle **SAS**.

EstructuraApiClient.Core

- Boundary\Contracts\Repositories

IAreaHospitalariaRepository

```
AreaHospitalaria GetAreaHospitalariaByCodigo(string codigo);
AreaHospitalaria GetAreaHospitalariaByUnidadFuncional(string codigo);
AreaHospitalaria ObtenerAreaHospitalariaReferencia(decimal codigo, string
codigoEspecialidad);
IList<AreaHospitalaria> GetAreasHospitalariasByProvincia(string codigo);
IList<AreaHospitalaria> GetAreasHospitalariasReferenciaByAreaHospitalaria
(string codigo);
```

ICentroEspecializadaRepository

```
CentroEspecializada GetCentroEspecializadaByCodigo(string
codigo);
IList<CentroEspecializada> GetCentrosEspecializadaByUnidadFuncional(string
codigo);
```

ICentroPrimariaRepository


```
        CentroPrimaria GetCentroPrimariaByCodigo(string codigo);  
        IList<CentroPrimaria> GetCentrosPrimariaByZonaBasica(string codigo);
```

ICentroPrivadoRepository

```
        CentroPrivado GetCentroPrivadoByCodigo(string codigo);  
        IList<CentroPrivado> GetCentrosPrivadosByProvincia(string codigo);
```

IDiagnosticoCieRepository

```
        DiagnosticoCie GetDiagnosticoCieByCodigo(string codigo);  
        IList<DiagnosticoCie> GetDiagnosticoCieByCodigoLike(string codigo);  
        IList<DiagnosticoCie> GetDiagnosticoCieByDescripcion(string descripcion);  
        IList<DiagnosticoCie> GetDiagnosticoCieByCodigoDescripcion(string codigo,  
        string descripcion);
```

IDispositivoApoyoRepository

```
        DispositivoApoyo GetDispositivoApoyoByCodigo(string codigo);  
        IList<DispositivoApoyo> GetDispositivosApoyoByDistrito(string codigo);
```

IDistritoRepository

```
        Distrito GetDistritoByCodigo(string codigo);  
        Distrito GetDistritoByDispositivoApoyo(string codigo);  
        Distrito GetDistritoByCentroPrimaria(string codigo);  
        IList<Distrito> GetDistritosByProvincia(string codigo);
```

IEspecialidadRepository

```
        Especialidad GetEspecialidadByCodigo(string codigo);
Especialidad GetEspecialidadByUnidadFuncional(string codigo);
IList<Especialidad> GetEspecialidades();
```

ISexoRepository

```
        Sexo GetSexoByCodigo(string codigo);
IList<Sexo> GetSexos();
```

ITareaRepository

```
        Tarea GetTareaByCodigo(string codigo);
IList<Tarea> GetTareasByRequiereSolicitud(string requiereSolicitud);
IList<Tarea> GetTareasByTipoAsistencia(string requiereSolicitud, string
tipoAsistencia);
IList<Tarea> GetTareasByTipoAsistenciaProcedeAp(string requiereSolicitud,
string tipoAsistencia, string procedeAp);
```

ITipoAsistenciaRepository

```
        TipoAsistencia GetTipoAsistenciaByCodigo(string codigo);
IList<TipoAsistencia> GetTiposAsistenciasByUnidadFuncionalCentroFisico
(string unidadFuncionalCodigo, string centroFisicoCodigo);
IList<TipoAsistencia> GetTiposAsistencia();
IList<TipoAsistencia> GetTiposAsistenciasByCentroPrimaria(string codigo);
IList<TipoAsistencia> GetTiposAsistenciasByDispositivoApoyo(string codigo);
```

IUbicacionRepository

```
        Ubicacion GetUbicacionByCodigo(string codigo);
IList<Ubicacion> GetUbicacionesByTipo(string tipo);
```

IAreaHospitalariaService

```
        AreaHospitalaria GetAreaHospitalariaByCodigo
(IGenericAuthContext<object> context, string codigo);
        AreaHospitalaria GetAreaHospitalariaByUnidadFuncional
(IGenericAuthContext<object> context, string codigo);
        AreaHospitalaria ObtenerAreaHospitalariaReferencia
(IGenericAuthContext<object> context, decimal codigo, string
codigoEspecialidad);
        IList<AreaHospitalaria> GetAreasHospitalariasByProvincia
(IGenericAuthContext<object> context, string codigo);
        IList<AreaHospitalaria> GetAreasHospitalariasReferenciaByAreaHospitalaria
(IGenericAuthContext<object> context, string codigo);
```

ICentroEspecializadaService

```
        CentroEspecializada GetCentroEspecializadaByCodigo
(IGenericAuthContext<object> context, string codigo);
        IList<CentroEspecializada> GetCentrosEspecializadaByUnidadFuncional
(IGenericAuthContext<object> context, string codigo);
```

ICentroPrimariaService

```
        CentroPrimaria GetCentroPrimariaByCodigo
(IGenericAuthContext<object> context, string codigo);
        IList<CentroPrimaria> GetCentrosPrimariaByZonaBasica
(IGenericAuthContext<object> context, string codigo);
```

ICentroPrivadoService

```
        CentroPrivado GetCentroPrivadoByCodigo
(IGenericAuthContext<object> context, string codigo);
        IList<CentroPrivado> GetCentrosPrivadosByProvincia
(IGenericAuthContext<object> context, string codigo);
```

IDiagnosticoCieService

```
        DiagnosticoCie GetDiagnosticoCieByCodigo
(IGenericAuthContext<object> context, string codigo);
        IList<DiagnosticoCie> GetDiagnosticoCieByCodigoLike
(IGenericAuthContext<object> context, string codigo);
        IList<DiagnosticoCie> GetDiagnosticoCieByDescripcion
(IGenericAuthContext<object> context, string descripcion);
        IList<DiagnosticoCie> GetDiagnosticoCieByCodigoDescripcion
(IGenericAuthContext<object> context, string codigo, string descripcion);
```

IDispositivoApoyoService

```
        DispositivoApoyo GetDispositivoApoyoByCodigo
(IGenericAuthContext<object> context, string codigo);
        IList<DispositivoApoyo> GetDispositivosApoyoByDistrito
(IGenericAuthContext<object> context, string codigo);
```

IDistritoService

```
        Distrito GetDistritoByCodigo(IGenericAuthContext<object>
context, string codigo);
        Distrito GetDistritoByDispositivoApoyo(IGenericAuthContext<object> context,
string codigo);
        Distrito GetDistritoByCentroPrimaria(IGenericAuthContext<object> context,
string codigo);
        IList<Distrito> GetDistritosByProvincia(IGenericAuthContext<object> context,
string codigo);
```

IEspecialidadService

```
        Especialidad GetEspecialidadByCodigo
(IGenericAuthContext<object> context, string codigo);
        Especialidad GetEspecialidadByUnidadFuncional(IGenericAuthContext<object>
context, string codigo);
        IList<Especialidad> GetEspecialidades(IGenericAuthContext<object> context);
```

ISexoService

```
        Sexo GetSexoByCodigo(IGenericAuthContext<object> context,
string codigo);
        IList<Sexo> GetSexos(IGenericAuthContext<object> context);
```

ITareaService

```
        Tarea GetTareaByCodigo(IGenericAuthContext<object> context,
string codigo);
        IList<Tarea> GetTareasByRequiereSolicitud(IGenericAuthContext<object>
context, string requiereSolicitud);
        IList<Tarea> GetTareasByTipoAsistencia(IGenericAuthContext<object> context,
string requiereSolicitud, string tipoAsistencia);
        IList<Tarea> GetTareasByTipoAsistenciaProcedeAp(IGenericAuthContext<object>
context, string requiereSolicitud, string tipoAsistencia, string procedeAp);
```

ITipoAsistenciaService

```

        TipoAsistencia GetTipoAsistenciaByCodigo
(IGenericAuthContext<object> context, string codigo);
        IList<TipoAsistencia> GetTiposAsistenciasByUnidadFuncionalCentroFisico
(IGenericAuthContext<object> context, string unidadFuncionalCodigo, string
centroFisicoCodigo);
        IList<TipoAsistencia> GetTiposAsistencia(IGenericAuthContext<object>
context);
        IList<TipoAsistencia> GetTiposAsistenciasByCentroPrimaria
(IGenericAuthContext<object> context, string codigo);
        IList<TipoAsistencia> GetTiposAsistenciasByDispositivoApoyo
(IGenericAuthContext<object> context, string codigo);

```

IUbicacionService

```

        Ubicacion GetUbicacionByCodigo(IGenericAuthContext<object>
context, string codigo);
        IList<Ubicacion> GetUbicacionesByTipo(IGenericAuthContext<object> context,
TipoUbicaciones tipo);

```

IUnidadFuncionalEspecializadaService

```

        UnidadFuncional GetUnidadFuncionalByCodigo
(IGenericAuthContext<object> context, string codigo);
        UnidadFuncional GetLaboratorioAreaHospitalaria(IGenericAuthContext<object>
context, string codigoAreaHospitalaria);
        IList<UnidadFuncional> GetUnidadesFuncionalesByAreaHospitalariaEspecialidad
(IGenericAuthContext<object> context, string codigoAreaHospitalaria, string
codEspecialidad);
        IList<UnidadFuncional>
GetUnidadesFuncionalesByAreaHospitalariaEspecialidadAgendasPrimaria
(IGenericAuthContext<object> context, string codigoAreaHospitalaria, string
codEspecialidad, string agendasPrimaria);
        IList<UnidadFuncional> GetUnidadesFuncionalesByAreaHospitalaria
(IGenericAuthContext<object> context, string codigoAreaHospitalaria);

```

IZonaBasicaService

```

        ZonaBasica GetZonaBasicaByCodigo(IGenericAuthContext<object>
context, string codigo);
        IList<ZonaBasica> GetZonasBasicasByDistrito(IGenericAuthContext<object>
context, string codigo);
        ZonaBasica GetZonaBasicaByCentroFisicoAtencionPrimaria
(IGenericAuthContext<object> context, string codigo);

```

IEstructuraFhirMapper

```

        LocalizacionesReferentes MapLocalizacionesReferentes(string
resource);

```

- Control\Utility

DescriptionAttributeExtension

```

        public static string GetStringValue(this Enum value)

```

StringValueAttribute

- Entity

AreaHospitalaria (Codigo, Descripcion, DescripcionOriginal, AhReferencia, Privada, Orden)
AreasHospitalarias
CentroEspecializada (Codigo, Descripcion)
CentroPrimaria (Codigo, Descripcion)
CentroPrivado (Codigo, Descripcion, DescripcionCorta)
CentrosEspecializados
CentrosPrimarios
CentrosPrivados
DiagnosticoCie (Codigo, Descripcion)
DiagnosticosCie
DispositivoApoyo (Codigo, Descripcion, Orden)
DispositivosApoyo
Distrito (Codigo, Descripcion)
Distritos
Enumerados
 TipoUnidad: Primaria, Especializada
Especialidad (Codigo, Descripcion, Decreto, Primaria)
Especialidades
LocalizacionesReferentes
MacoContext
Modulo (Codigo, Descripcion)
PerfilModulo (Codigo, Descripcion)
Profesional (Codigo, Nombre, Apellido1, Apellido2, CNP)
Provincia (Codigo, Descripcion)
Sexo (Codigo, Descripcion)
Sexos
Tarea (Codigo, Descripcion, DescripcionCorta, Decreto, ProcedePrimaria)
Tareas
TipoAsistencia (Codigo, Descripcion, Ambito, Decreto)
TipoLocalizacionFisica: Enumerado con valores: AreaHospitalaria, CentroAtencionPrimaria, DispositivoApoyo
TiposAsistencia
TipoUbicaciones. Enumerado con los distintos tipos de ubicaciones: Admision, Archivo, Bloque_Area_Planta, etc

Ubicacion (Codigo,Descripcion,CodigoPadre)

Ubicaciones

UnidadesFuncionales

UnidadFuncional (Codigo, Descripcion)

ZonaBasica (Codigo, Descripcion)

ZonasBasicas

SAS.EstructuraApiClient

- Boundary\Contracts\Services

AreaHospitalariaService: Implementa la interfaz IAreaHospitalariaService

CentroEspecializadaService: Implementa la interfaz ICentroEspecializadaService

CentroPrimariaService: Implementa la interfaz ICentroPrimariaService

CentroPrivadoService: Implementa la interfaz ICentroPrivadoService

DiagnosticoCieService: Implementa la interfaz IDiagnosticoCieService

DispositivoApoyoService: Implementa la interfaz IDispositivoApoyoService

DistritoService: Implementa la interfaz IDistritoService

EspecialidadService: Implementa la interfaz IEspecialidadService

LocalizacionesReferentesService: Implementa la interfaz
ILocalizacionesReferentesService

MunicipioService: Implementa la interfaz IMunicipioService

ProvinciaService: Implementa la interfaz IProvinciaService

SexoService: Implementa la interfaz ISexoService

TareaService: Implementa la interfaz ITareaService

TipoAsistenciaService: Implementa la interfaz ITipoAsistenciaService

UbicacionService: Implementa la interfaz IUbicacionService

UnidadFuncionalEspecializadaService: Implementa la interfaz
IUnidadFuncionalEspecializadaService

ZonaBasicaService: Implementa la interfaz IZonaBasicaService

SAS.EstructuraApiClient.ManagedDataAccess.Oracle

- Boundary\

AreaHospitalariaRepository. Implementa la interfaz
 IAreaHospitalariaRepository
 CentroEspecializadaRepository. Implementa la interfaz ICentroEspecializadaRepository
 CentroPrimariaRepository. Implementa la interfaz ICentroPrimariaRepository
 CentroPrivadoRepository. Implementa la interfaz ICentroPrivadoRepository
 ConfigContextApiEstructuraConnection. Implementa la interfaz
 IConfigContextApiConnection
 DiagnosticoCieRepository. Implementa la interfaz IDiagnosticoCieRepository
 DispositivoApoyoRepository. Implementa la interfaz IDispositivoApoyoRepository
 DistritoRepository. Implementa la interfaz IDistritoRepository
 EspecialidadRepository. Implementa la interfaz IEspecialidadRepository
 ManagedConfigEstructuraConnection. Implementa la interfaz
 ManagedConfigConnectionBase
 MunicipioRepository. Implementa la interfaz IMunicipioRepository
 ProvinciaRepository. Implementa la interfaz IProvinciaRepository
 SexoRepository. Implementa la interfaz ISexoRepository
 TareaRepository. Implementa la interfaz ITareaRepository
 TipoAsistenciaRepository. Implementa la interfaz ITipoAsistenciaRepository
 UbicacionRepository. Implementa la interfaz IUbicacionRepository
 UnidadFuncionalRepository. Implementa la interfaz IUnidadFuncionalRepository
 ZonaBasicaRepository. Implementa la interfaz IZonaBasicaRepository

- Control\Mapper
 Conjunto de clases que mapean la información a las entidades del sistema
- Entity\EntityFramework BDU: Entidades del sistema BDU Especializada: Entidades del sistema Estructura
- Entity\Exceptions ContextException. Contexto para las excepcionescontext
- Entity\Projection Entidades proyección utilizadas en el sistema
- 1.4.0 - Posibilidad de contexto no manejado Se modifica la clase ManagedConfigConnectionBase añadiendo el atributo **IsManagedContext**. True para el caso de que el contexto sea manejado, False en caso contrario. Por defecto es False.
- 1.5.0 - Se añade método para consultar provincias **SAS.EstructuraApiClient.Core**
 - Boundary\Contracts\Repositories

IProvinciaRepository

```

        Provincia GetProvinciaBycentroFisicoAtencionPrimaria(string
codigo);
        Provincia GetProvinciaByDispositivoApoyo(string codigo);
        Provincia GetProvinciaByUnidadFuncional(string codigo);
        Provincia GetProvinciaByCodigo(string codigo);

```

- Boundary\Contracts\Services

IProvinciaService

```

        Provincia GetProvinciaBycentroFisicoAtencionPrimaria
(IGenericAuthContext<object> context, string codigo);
        Provincia GetProvinciaByDispositivoApoyo(IGenericAuthContext<object>
context, string codigo);
        Provincia GetProvinciaByCodigo(IGenericAuthContext<object> context, string
codigo);

```

- 1.6.0 - Modificar servicio de Provincias para que se pueda consultar por código de comunidad **SAS.EstructuraApiClient.Core**

- Boundary\Contracts\Repositories

IProvinciaRepository

```

        IList<Provincia> GetProvincias(decimal? codigoComunidad);

```

- Boundary\Contracts\Services

IProvinciaService

```

        IList<Provincia> GetProvincias(IGenericAuthContext<object>
context, decimal? codigoComunidad);

```

- 1.7.0 - Servicio que proporcione Localizaciones Preferentes, utilizando cliente http para llamar a servicio REST de estructura **SAS.EstructuraApiClient.Core**

- Boundary\Contracts\Repositories

ILocalizacionesReferentesRepository

```
Task<LocalizacionesReferentes> GetLocalizacionesReferentes
(MacoContext context, string id, TipoLocalizacionFisica tipoLocalizacion,
string type =null, int count=50, int page =1 );
```

- Boundary\Contracts\Services

ILocalizacionesReferentesService

```
Task<LocalizacionesReferentes> GetLocalizacionesReferentes
(MacoContext context, string id, TipoLocalizacionFisica tipoLocalizacion,
string type = null, int count = 50, int page = 1);
```

SAS.EstructuraApiClient.Fhir.R4

- Control\Mapper EstructuraFhirR4Mapper LocalizacionesReferentes MapLocalizacionesReferentes(string resourceBundle) **SAS.EstructuraApiClient.REST**
- \Boundary **ConfigHttpClientEstructura**. Configuración de conexiones al servicio REST **LocalizacionesReferentesRepository**. public async Task GetLocalizacionesReferentes(MacoContext context, string id, TipoLocalizacionFisica tipoLocalizacion, string type = null, int count = 50, int page = 1)
- \Control **EstructuraFhirMapperFactory**. Clase abstracta de la factoría mapper de entidades **EstructuraFhirMapperManaged**. Manejador de mapper de fhirs de estructura **EstructuraFhirR4MapperFactory**: Factoría mapper de la versión R4 de Fhir
- \Entity **EnumFhirVersions**. Enumerados de versiones de Fhir: R4 y STU3 **RequestUriLocationTypeEnum**. Enumerado con los tipos de peticiones Uri. Actualmente LocalizacionesReferentes.
- 1.7.1 - Modificación de los tipos Number en las entidades de BBDD En las clases de las entidades SAS. EstructuraApiClient.ManagedDataAccess.Entity se modifican tipos de atributos: de int? a decimal?, de short a int, de decimal? a bool?, etc.
- 1.8.0 - Se deshace la "interiorización" del componente hl7.fhir.r4 Se elimina de la librería que internamente contenga el componente hl7.fhir.r4
- 1.8.1 - Se elimina dependencia con Newtonsoft.Json en el fichero de configuración nuget
- 1.9.0 - Se añaden nuevos métodos de obtención de municipios, área hospitalaria de referencia **SAS.EstructuraApiClient.Core**

- Boundary\Contracts\Repositories

IAreaHospitalariaRepository

```

        IList<AreaHospitalaria> ObtenerAreasHospitalariasDestino
(string codigoCentro, string codigoEspecialidad, Enumerados.TipoUnidad
tipoUnidad);

```

IMunicipioRepository

```

        IList<Municipio> GetMunicipiosByProvincia(decimal?
codigoProvincia);

```

- Boundary\Contracts\Services

IAreaHospitalariaService

```

        IList<AreaHospitalaria> ObtenerAreasHospitalariasDestino
(IGenericAuthContext<object> context, string codigo, string
codigoEspecialidad, Enumerados.TipoUnidad tipoUnidad);

```

- **IMunicipioService**

```

        IList<Municipio> GetMunicipiosByProvincia
(IGenericAuthContext<object> context, decimal? codigoProvincia);

```

- Entity Municipio (Codigo,Descripcion)
- 1.9.1 - Actualización de las versiones de componentes: SAS.MacoApiClient a la versión 2.4.0 y SAS.Common.Core a la v1.1
- 1.10.0 - Creación de nuevo método getMunicipioByCodigo y de enumerados: Tipo Asistencia de la tarea e indicador de si la tarea procede de atención primaria **SAS.EstructuraApiClient.Core**

- Boundary\Contracts\Repositories **IMunicipioRepository.cs**. Se añade el siguiente método:

```

        Municipio GetMunicipioByCodigo(decimal codigoMunicipio);

```

ITareaRepository.cs. Se hacen los siguientes cambios

`IList<Tarea> GetTareasByRequiereSolicitud(bool requiereSolicitud);` El parámetro `requiereSolicitud` pasa de string a bool

`IList<Tarea> GetTareasByTipoAsistencia(bool requiereSolicitud, tipoAsistencia tipoAsistencia);` El parámetro `requiereSolicitud` pasa de string a bool y `tipoAsistencia` pasa de string a enumerado `TipoAsistencia`

`IList<Tarea> GetTareasByTipoAsistenciaProcedeAp(bool requiereSolicitud, tipoAsistencia tipoAsistencia, tipoProcedeAp procedeAp);` El parámetro `requiereSolicitud` pasa de string a bool, `tipoAsistencia` pasa de string a enumerado `TipoAsistencia` y `ProcedeAp` pasa de string a de tipo enumerado `TipoProcedeAP`

- Entity **Enumerados.cs**: Se han añadido los siguientes enumerados: `TipoAsistencia`: `AsistenciaPrimaria`, `Hospitalizacion`, `ProcedimientoDiagnosticoTerapeuticoNoQuirurgico`, `ConsultaExterna`, `Urgencia`, `HospitalDiaMedico`, `IntervencionQuirurgicaGiagnosticoTerapeutica`, `Laboratorio`, `HospitalDiaQuirurgico`, `UrgenciasFueraCentro`

`TipoProcedeAp`: `ConsultaExterna`, `Primaria`, `Hospitalizacion`, `Urgencias`, `ComunTodasEspecializada`

SAS.EstructuraApiClient.ManagedDataAccess.Oracle

- Boundary **MunicipioRepository.cs**. Se actualiza con los cambios definidos en la interfaz `IMunicipioRepository`
- Boundary **TareaRepository.cs**. Se actualiza con los cambios definidos en la interfaz `ITareaRepository`
- 1.11.0 - Se incluye un nuevo método para obtener áreas hospitalarias concertadas por su código de provincia y código de especialidad.

SAS.EstructuraApiClient.Core

- Boundary\Contracts\Repositories **IAreaHospitalariaRepository.cs**. Se añade el siguiente método:

`IList<AreaHospitalaria>`

`GetAreasHospitalariasConcertadasByProvinciaEspecialidad(string codigoProvincia, string codigoEspecialidad);`

- Boundary\Contracts\Services **IAreaHospitalariaService.cs**. Se añade el siguiente método:

```

        IList<AreaHospitalaria>
        GetAreasHospitalariasConcertadasByProvinciaEspecialidad
        (IGenericAuthContext<object> context, string codigoProvincia, string
        codigoEspecialidad);

```

- 1.12.0 - Se incluye un nuevo método para obtener los centros privados mediante el código de un área hospitalaria y el código de la unidad funcional.

SAS.EstructuraApiClient.Core

- Boundary\Contracts\Repositories **ICentroPrivadoRepository.cs**. Se añade el siguiente método:

```

        IList<CentroPrivado>
        GetCentrosPrivadosByAreaHospitalariaUnidadFuncional(string
        codigoAreaHospitalaria, string codigoUnidadFuncional);

```

- Boundary\Contracts\Services **ICentroPrivadoService.cs**. Se añade el siguiente método:

```

        IList<CentroPrivado>
        GetCentrosPrivadosByAreaHospitalariaUnidadFuncional
        (IGenericAuthContext<object> context, string codigoAreaHospitalaria, string
        codigoUnidadFuncional);

```

- 1.12.1 - Se actualiza la versión de los componentes: SAS.Common.Oracle a v1.3.3, SAS.MacoApiClient v2.4.1, debido a que internamente se ha cambiado las librerías Oracle.ManagedDataAccess y Oracle.ManagedDataAccess.EntityFramework a la versión 19.6.0
- 1.13.0 - Se incluyen OrderBy por el campo description a consultas que devuelven listado de entidades con dicho atributo. También se Modifican las consultas del tipo GetXXXByCodigo(string codigo) para que muestren código y descripción
- 1.13.1 - Se regulariza versiones de componentes:SAS.Common.Core se actualiza a la v1.1.0 y SAS.MacoApiClient a v2.4.2
- 1.13.2 - Se regulariza la versión del componentes SAS.MacoApiClient a v2.4.3, la cual permite comunicación con servicios Maco usando HTTPS
- 1.13.3 - Homogeneizar las consultas de especialidad y tipo de asistencia. **SAS.EstructuraApiClient.ManagedDataAccess.Oracle**

- Boundary\Repository **TipoAsistenciaRepository.cs**. Se modifica el método GetEspecialidadByUnidadFuncional() para recuperar el código de la especialidad, y en el método GetTipoAsistenciaByCodigo() también se obtiene el ámbito y decreto del tipo de asistencia.

- 1.13.4 - Se corrige error en el método GetDistritoByCentroPrimaria. Se modifica las tablas orígenes de donde se accede a la información **SAS.EstructuraApiClient.ManagedDataAccess.Oracle**
 - Boundary\Repository **DistritoRepository.cs**. Ahora se obtienen los datos por Centros_AP, Zona_Basica_Salud y Distrito

Integración de la librería

Configurar en Microsoft Visual Studio un nuevo origen de paquetes Nuget con la dirección de los artefactos .NET del SAS. Para ello, seguir las indicaciones de la página [Repositorio de artefactos](#). Por último, instalar la librería SAS.EstructuraApiClient.

Configuración

Debido a la versión de EntityFramework y del componente proveedor de BBDD Oracle, la librería requiere el fichero de configuración .config.

- En el fichero de configuración de la aplicación, **app.config**, se define la cadena de conexión de la BBDD de Estructura.
- En el fichero **Startup.cs**, método **ConfigureServices**, se inicializa el contenedor de inyección de dependencia

```
public void ConfigureServices(IServiceCollection services)
```

```
{
```

```
...
```

```
    ContainerDI.SetAsyncScopedLifestyle(); // establecer estilo de vida por defecto del contenedor de
    inyección. AsyncScopedLifestyle 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.

- En el fichero **Startup.cs**, método **InitializeContainer**, se ha definido las instrucciones necesarias para configurar el componente de Estructura, así como la inyección de dependencia de los servicios/repositorios utilizados en éste componente.

Versión <= 1.1.2

```

        IConfigContextApiConnection config = new ConfigContextApiConnection
    (
        new ManagedConfigConnection[]
        {
            new ManagedConfigConnection("DATA SOURCE=xxxxx; PASSWORD = xxxxx; PERSIST SECURITY INFO = True;
USER ID = xxxxx", SchemaType.Especializada, "REP_PRO_ESPECIALIZADA", logInfo => logger.Log(logInfo)),
            new ManagedConfigConnection("DATA SOURCE=xxxxx; PASSWORD = xxxxx; PERSIST SECURITY INFO =
True; USER ID = xxxxx", SchemaType.Maco, "REP_PRO_MACO", logInfo => logger.Log(logInfo))
        }, true
    );

```

Versión 1.2

```

        IConfigContextApiConnection<SchemaType> = this.config = new ConfigContextApiEstructuraConnection
    (
        new ManagedConfigEstructuraConnection[]
        {
            new ManagedConfigEstructuraConnection("ConnectionStringEspecializadaTest",
SchemaType.Especializada, "REP_PRO_ESPECIALIZADA"),
            new ManagedConfigEstructuraConnection("ConnectionStringBduTest", SchemaType.Bdu,
"REP_PRO_BDU")
        }, false
    );

```

Versión >= 1.3

```

        ContainerDI.Register<IConfigContextApiConnection<SchemaEstructuraType>>(() =>
new ConfigContextApiEstructuraConnection(new ManagedConfigConnectionBase<SchemaEstructuraType>[]
{
    new ManagedConfigConnectionBase<SchemaEstructuraType>
("ConnectionStringEspecializadaTest", SchemaEstructuraType.Especializada, "REP_PRO_EST"),
    new ManagedConfigEstructuraConnection("ConnectionStringBduTest",
SchemaEstructuraType.Bdu, "REP_PRO_BDU")
})
, ContainerDI.LifeStyleContainer.Transient);

        ContainerDI.Register<ICentroPrivadoRepository, CentroPrivadoRepository>(ContainerDI.
LifeStyleContainer.Transient);
        ContainerDI.Register<ICentroPrivadoService<object>, CentroPrivadoService>(ContainerDI.
LifeStyleContainer.Transient);
        ContainerDI.Register<IEstructuraService, EstructuraService>(ContainerDI.LifeStyleContainer.
Transient);
        ContainerDI.Register<IAreaHospitalariaRepository, AreaHospitalariaRepository>(ContainerDI.
LifeStyleContainer.Transient);
        ContainerDI.Register<IAreaHospitalariaService<object>, AreaHospitalariaService>(ContainerDI.
LifeStyleContainer.Transient);
        ContainerDI.Register<IAreaHospitalariaUseCase, AreaHospitalariaUseCase>(ContainerDI.
LifeStyleContainer.Transient);
        ContainerDI.Register<ICentrosPrivadosUseCase, CentrosPrivadosUseCase>(ContainerDI.LifeStyleContainer.
Transient);

```

Uso

Definir en el constructor de la clase donde se desea utilizar servicios de estructura la interfaz de ésta que se desee utilizar. Por ejemplo, para el servicio de Centros Privados:

```
public EstructuraService(ICentroPrivadoService<object> centroPrivadoService, ...)  
{  
    _centroPrivadoService = centroPrivadoService;  
    ...  
}
```

Llamar al método de la interfaz instanciada que corresponda pasándole los parámetros adecuados.