



G VICEPRESIDÈNCIA
O I CONSELLERIA
I INNOVACIÓ,
B RECERCA I TURISME
/ DIRECCIÓ GENERAL
DESENVOLUPAMENT
TECNOLÒGIC

una manera de hacer
europa 

Fondo Europeo de
Desarrollo Regional



Unió Europea

DISTRIBUCIO: Manual d'integració

10 de maig de 2023

Serveis d'Administració Electrònica en el Govern de les Illes Balears

Lot 3 (Serveis de tramitació d'expedients electrònics)

Oficina Tècnica de Direcció de Projecte

Control de versions del document

Control de Canvis			
Data	Autor	Versió	Canvis
6.10.18	Límit Technologies	1.0	Versió inicial
3.12.18	Límit Technologies	1.1	Modificació del document segons plantilla enviada per la DGDT.
31/10/19	Límit Technologies	1.2	Nova llibreria d'utilitats per a backoffices de Distribució.
24/06/20	Límit Technologies	1.3	Ampliació llibreria d'utilitats per backoffices de Distribució.
10/05/23	Límit Technologies	1.4	Afegir punts per explicar les APIs REST d'integració per backoffices i per crear regles de distribució de forma remota.

Revisat per		
Nom	Data	Àrea, departament o empresa

Aprovat per		
Nom	Data	Àrea, departament o empresa

Llista de distribució		
Nom	Àrea, departament o empresa	Correu electrònic

Índex

Control de versions del document.....	2
1. Objecte.....	5
2. Enviament d'anotacions a DISTRIBUCIO.....	5
2.1. Diagrama de funcionament.....	5
2.1.1. enviarAnotacioRegistreEntrada (síncron).....	5
2.1.2. enviarAnotacioRegistreEntrada (asíncron).....	6
2.2. Model de classes.....	7
2.3. Descripció del servei.....	13
2.4. Consideracions de disseny i ús de l'API.....	13
2.4.1. Adreça del servei.....	13
2.4.2. Segurització del servei.....	13
2.4.3. Dependències Maven.....	13
2.5. Client d'exemple.....	14
3. Backoffices de DISTRIBUCIO.....	20
3.1. Distribucio d'anotacions cap a backoffices.....	20
3.1.1. Integració com a backoffice de DISTRIBUCIO.....	21
3.1.2. WS d'integració amb els backoffices.....	21
3.1.3. API REST d'integració amb backoffice.....	26
3.1.4. API REST de creació de regles.....	31
3.2. Mètode Consulta.....	36
3.3. Mètode CanviEstat.....	36
3.4. Diagrama de funcionament.....	37
3.5. Model de dades.....	38
4. Llibreria d'utilitats pels backoffices de DISTRIBUCIO.....	38
4.1. Informació de la llibreria.....	38
4.2. Com incloure la llibreria.....	39
4.3. Històric.....	40
4.4. Utilitats per l'Arxiu.....	40
4.4.1. Paràmetres.....	41
4.4.2. Ús de <i>BackofficeArxiuUtils</i>	42
4.4.3. Exemple Java.....	43
4.5. Utilitats per Sistra2.....	47
4.5.1. Ús de <i>BackofficeSistra2Utils</i>	48
4.5.2. Exemple Java.....	49
Annex I: WSDL del servei d'enviament a bústies.....	50
Annex II: WSDL del backoffice.....	58
Annex III: WSDL del servei d'integració.....	60
Annex IV: Exemple d'ús de la llibreria de <i>distribucio-backoffice-utils</i>.....	68

1. Objecte

En aquest document es proporciona tota la informació necessària per a la integració d'una aplicació amb els serveis de DISTRIBUCIÓ.

A l'apartat 2 es detalla com integrar una aplicació per enviar anotacions de registre a DISTRIBUCIO.

A l'apartat 3 es detalla com integrar una aplicació per rebre anotacions de registre de DISTRIBUCIO com a backoffice de DISTRIBUCIO per processar i informar del seu estat.

2. Enviament d'anotacions a DISTRIBUCIO

La integració amb DISTRIBUCIO per enviar anotacions de registre

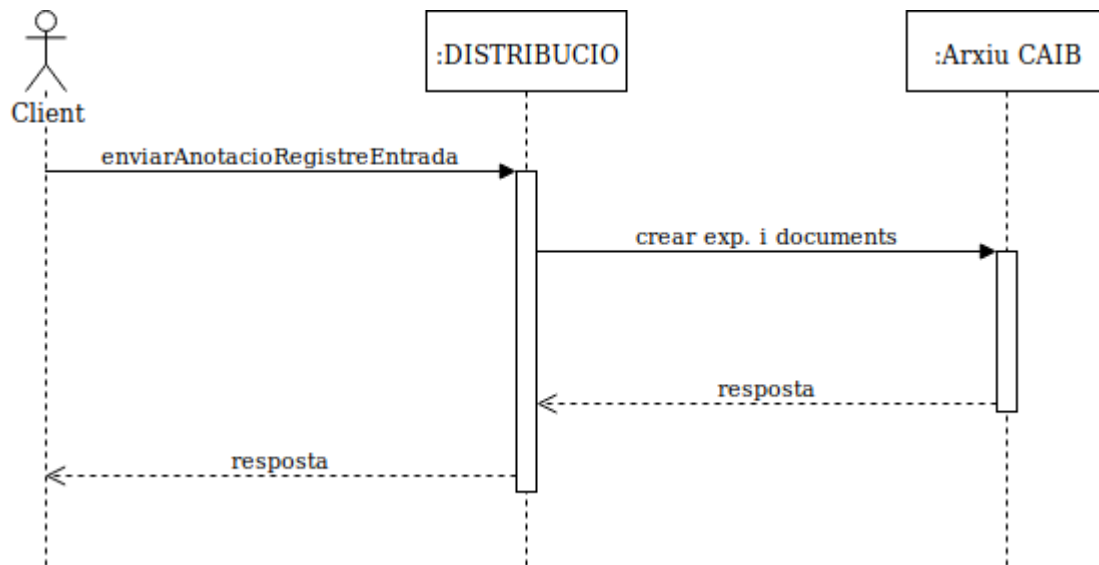
2.1. Diagrama de funcionament

El servei d'enviament a bústies permet l'enviament d'una anotació de registre a una de les bústies gestionades per DISTRIBUCIO. Una vegada l'anotació arriba a DISTRIBUCIO, aquesta es processa segons les regles definides a l'aplicació i es donen d'alta els documents annexos a dins l'arxiu digital de la CAIB.

Segons la configuració de l'aplicació, el processament de l'anotació de registre es pot dur a terme de forma síncrona o de forma asíncrona.

2.1.1. enviarAnotacioRegistreEntrada (síncron)

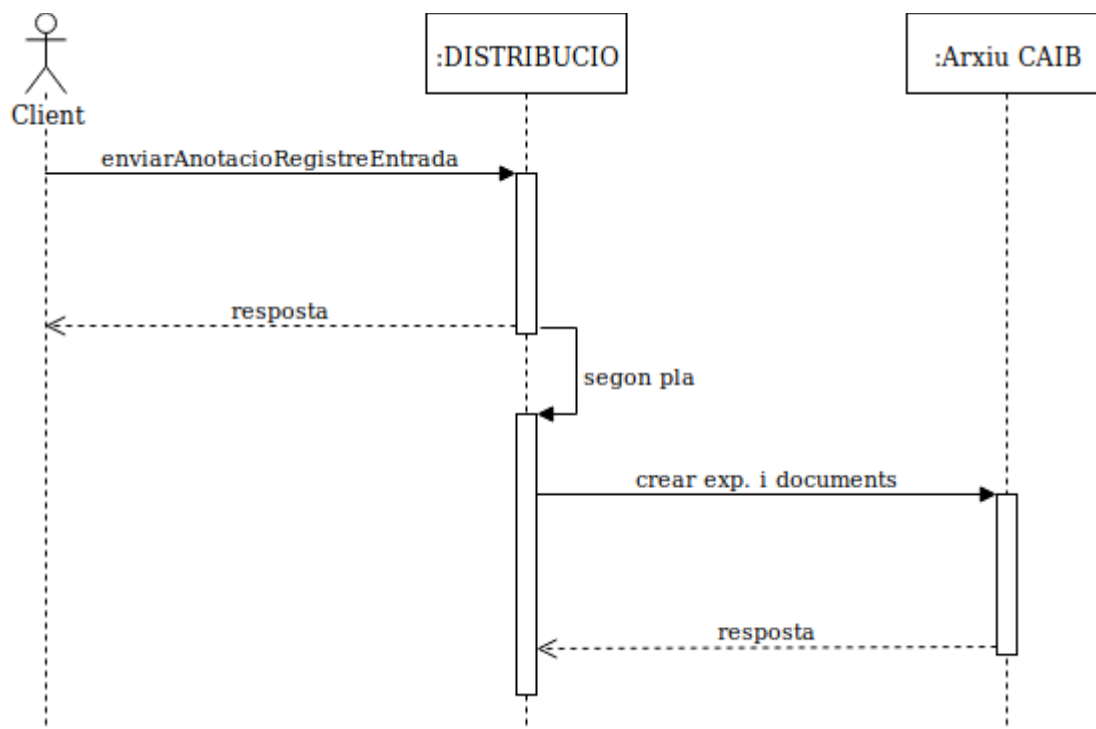
En aquest cas d'ús el processament de l'anotació de registre es farà de forma síncrona. És a dir, que fins que l'anotació no s'hagi processat correctament, i els annexos de l'anotació de registre s'hagin guardat correctament a l'arxiu, no es retornarà una resposta al client.



La resposta retornada al client podrà retornar un error si alguna de les passes efectuades no s'ha pogut completar correctament.

2.1.2. enviarAnotacioRegistreEntrada (asíncron)

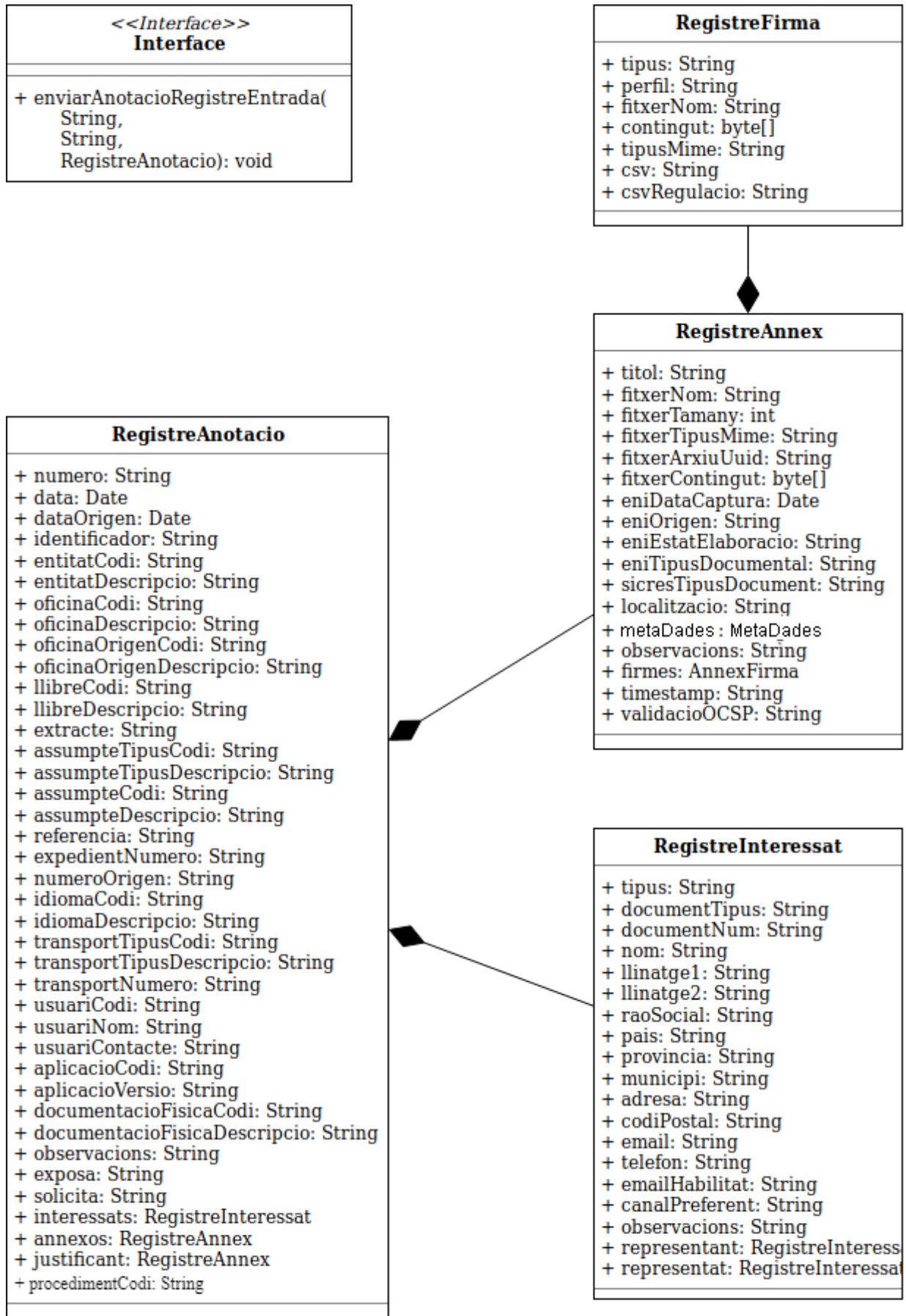
En aquest cas d'us el processament de l'anotació de registre es farà de forma asíncrona. És a dir, que l'anotació es guardarà a dins distribució i es retornarà immediatament una resposta al client. Posteriorment, un procés en segon pla anirà processant les anotacions de registre pendent i guardarà els annexos associats a dins l'arxiu.



La resposta retornada al client podrà retornar un error si no s'ha pogut emmagatzemar correctament l'anotació a dins la base de dades de DISTRIBUCIO. Però a diferència del processament síncron, el client no rebrà cap tipus de feedback si es produeix algun error durant el processament de l'anotació de registre.

2.2. Model de classes

Aquest és el model de classes del servei d'enviament d'anotacions de registre.



A continuació explicam amb més detall els camps de cada una de les classes del model de dades.

RegistreAnotacio				
Nom del camp	Tipus	Mult.	Tamany	Descripció
tipusES	Cadena	0..1	1	Tipus d'assentament registral: - "E": entrada - "S": sortida El camp no és obligatori i si no s'informa es prendrà "E" per defecte.
numero	Cadena	1	255	Número identificador de l'assentament registral. És únic.
data	Data	1	14	Data de l'assentament.
dataOrigen	Data	0..1	14	Data origen de l'assentament.
identificador	Cadena	1	100	Codi identificador de l'assentament registral.
entitatCodi	Cadena	0..1	255	Codi DIR3 de l'entitat governamental a la que va dirigit l'assentament registral.
entitatDescripcio	Cadena	0..1	255	Descripció DIR3 de l'entitat governamental a la que va dirigit l'assentament registral.
oficinaCodi	Cadena	1	21	Codi DIR3 de l'oficina on es registre.
oficinaDescripcio	Cadena	0..1	300	Denominació DIR3 de l'entitat registral on es registre l'assentament.
oficinaOrigenCodi	Cadena	0..1	21	
oficinaOrigenDescripcio	Cadena	0..1	100	
llibreCodi	Cadena	1	4	Codi del llibre al qual s'ha fet l'assentament registral rebut.
llibreDescripcio	Cadena	0..1	255	Denominació del llibre al qual s'ha fet l'assentament registral rebut.
extracte	Cadena	1	240	Extracte o resum de l'assentament.
assumpteTipusCodi	Cadena	0..1	16	Codi del tipus d'assumpte de l'assentament registral.
assumpteTipusDescripcio	Cadena	0..1	100	Denominació del tipus d'assumpte de l'assentament registral.
assumpteCodi	Cadena	0..1	16	Codi de l'assumpte.
assumpteDescripcio	Cadena	0..1	255	Denominació de l'assumpte.
referencia	Cadena	0..1	16	Referència externa a qualche element de l'assentament.
expedientNumero	Cadena	0..1	80	Referència al nombre d'expedient de

				l'assentament.
numeroOrigen	Cadena	0..1	80	
idiomaCodi	Cadena	1	2	Codi d'idioma amb el que s'ha fet l'assentament.
idiomaDescripcio	Cadena	0..1	100	Descripció d'idioma amb el que s'ha fet l'assentament.
transportTipusCodi	Cadena	0..1	20	Codi SICRES3 del segment internos – tipo transporte.
transportTipusDescripcio	Cadena	0..1	100	Descripció SICRES3 del segment internos – tipo transporte.
transportNumero	Cadena	0..1	20	Número del transport.
usuariCodi	Cadena	0..1	20	Codi de l'usuari que vol realitzar l'operació.
usuariNom	Cadena	0..1	767	Nom de l'usuari que vol realitzar l'operació.
usuariContacte	Cadena	0..1	255	Contacte de l'usuari que fa l'assentament. Si és un registre electrònic el responsable del tràmit.
aplicacioCodi	Cadena	0..1	255	Codi de l'aplicació que fa l'enviament de l'assentament registral.
aplicacioVersio	Cadena	0..1	255	Versió de l'aplicació que fa l'enviament de l'assentament registral.
documentacioFisicaCodi	Cadena	0..1	19	Codi SICRES del segment internos – Documentación física y/o soportes.
documentacioFisicaDescripcio	Cadena	0..1	100	Descripció del segment internos – Documentación física y/o soportes
observacions	Cadena	0..1	50	Observacions de l'assentament.
exposa	Cadena	0..1	lob	Exposició de fets.
solicita	Cadena	0..1	lob	Objecte de la sol·licitud si escau.
interessats	RegistreInteressat	0..n		Dades que componen els interessats de l'assentament registral si n'hi ha.
annexos	RegistreAnnex	0..n		Dades que componen els annexos dels assentaments registrals si n'hi ha.
justificant	RegistreAnnex	0..1		Dades que componen el justificant de l'assentament registral si n'hi ha. Es compon de les mateixes dades que un annex.
procedimentCodi	Cadena	0..1	64	Codi de procediment.
presencial	Boolea	0..1		Paràmetre booleà per indicar si el registre és presencial

RegistreAnnex

Nom del camp	Tipus	Mult.	Tamany	Descripció
titol	Cadena	1	200	Títol del document
fitxerNom	Cadena	1	256	Nom del fitxer annexat
fitxerTamany	Numèric	0..1		Tamany en bytes del fitxer annexat
fitxerTipusMime	Cadena	1	30	Tipus MIME del fitxer annexat
fitxerArxiuUuid	Cadena	0..1	256	Uuid de l'arxiu digital si s'escau
fitxerContingut	Byte[]	0..1		Cadena de bytes que compon el contingut del fitxer annexat
eniDataCaptura	Data	1	14	Data de captura del document
eniOrigen	Cadena	1	10	Codi de la metadada origen de la NTI de document electrònic
eniEstatElaboracio	Cadena	1	4	Codi de la metadada estat elaboració de la NTI de document electrònic.
eniTipusDocumental	Cadena	1	255	Codi de la metadada tipus documental de la NTI de document electrònic.
sicresTipusDocument	Cadena	0..1	2	Codi SICRES del segment Anexo – tipo de documento
localitzacio	Cadena	0..1	80	
metaDades	MetaDades	0..1		Camp MetaDada amb entrades de tipus codi i valor per metadades addicionals.
observacions	Cadena	0..1	50	Observacions a destacar sobre la informació annexada
firmes	Firma	0..1		Llista de firmes del fitxer
timestamp	Cadena	0..1	100	Segell de temps del fitxer annex en base64.
validacioOCSP	Cadena	0..1	255	Validació OCSP del certificat emprat per signar en base64.

RegistreFirma				
Nom del camp	Tipus	Mult.	Tamany	Descripció
tipus	Cadena	1	30	Codi del tipus de firma
perfil	Cadena	0..1	30	Codi de perfil de la firma.
fitxerNom	Cadena	0..1	256	Nom del fitxer de firma de l'annex si n'hi ha. És obligatori si el tipus de firma és TF04.
contingut	Byte[]	0..1	var	Cadena de bytes que componen el contingut del fitxer de firma annex si n'hi ha. És obligatori si el tipus de firma és TF04.

tipusMime	Cadena	0..1	30	Tipus MIME del fitxer de firma annexat si n'hi ha. És obligatori si el tipus de firma és TF04.
csv	Cadena	0..1		En cas de que el fitxer es signi amb CSV, contingut del CSV en base64.
csvRegulacio	Cadena	1	640	Text de regulació CSV.

RegistreInteressat				
Nom del camp	Tipus	Mult.	Tamany	Descripció
tipus	Cadena	1	19	Figura jurídica de la persona '1' = Administració. '2' = Persona física. '3' = Persona jurídica.
documentTipus	Cadena	0..1	1	Codi SICRES del segment interesado – documento de identificación de interesado tipo.
documentNum	Cadena	0..1	17	Número de document identificatiu.
nom	Cadena	C	255	Nom de l'interessat. Obligatori si és una persona física.
l·linatge1	Cadena	C	255	Primer l·linatge de l'interessat. Obligatori si és una persona física.
l·linatge2	Cadena	0..1	255	Segon l·linatge de l'interessat
raoSocial	Cadena	C	2000	Raó social de l'interessat. Obligatori si és una persona jurídica.
pais	Cadena	C	100	Codi ISO numèric del País de la persona.
provincia	Cadena	C	100	Codi INE de la província de la persona.
municipi	Cadena	C	100	Codi del Municipi de la persona.
adresa	Cadena	C	160	Adreça postal de l'interessat.
codiPostal	Cadena	C	5	
email	Cadena	0..1	160	
telefon	Cadena	0..1	20	
emailHabilitat	Cadena	0..1	160	En cas de disposar d'adreça electrònica habilitada.
canalPreferent	Cadena	0..1	2	Codi SICRES del segment interesado – canal preferente de notificación del interesado.
codiDire	cadena	0..1	20	Codi DIRe de l'interessat.
observacions	Cadena	0..1	160	
representant	RegistreInteressat	0..1		Interessat que actua de representant

				de l'actual.
representat	RegistreInteressat	0..1		Interessat que actua com a representat per l'actual.

2.3. Descripció del servei

El servei d'enviament a bústies permet enviar anotacions de registre a les bústies donades d'alta a l'aplicació de DISTRIBUCIO. Aquest servei disposa del mètode **enviarAnotacioRegistreEntrada** que es pot utilitzar per a enviar anotacions provinents del registre d'entrada.

El mètode **enviarAnotacioRegistreEntrada** requereix especificar els següents paràmetres:

- entitat: El codi de l'entitat de DISTRIBUCIO a on es vol donar d'alta l'anotació.
- unitatAdministrativa: El codi DIR3 de la unitat administrativa destinatària de l'anotació de registre.
- registreEntrada: objecte amb la informació associada a l'anotació de registre.

Per a més informació sobre com emplenar correctament l'objecte de tipus RegistreEntrada es pot consultar el punt 2.2 d'aquest mateix document.

Aquest mètode no retorna cap informació si tot el procés de donar d'alta l'anotació de registre a dins DISTRIBUCIÓ ha funcionat correctament. En canvi, si s'ha produït algun tipus d'error, aquest es retorna com un missatge de tipus SoapFault que conté la descripció de l'error.

2.4. Consideracions de disseny i ús de l'API

2.4.1. Adreça del servei

L'aplicació desplega aquest servei en la següent adreça:

<http://localhost:8080/distribucio/ws/v1/bustia>

Aquesta adreça és vàlida per a accedir a un servei desplegat a la màquina local. Per a accedir al servei publicat en una màquina remota s'haurà de substituir «localhost» pel nom del servidor remot i el port «8080» pel port del servidor remot.

2.4.2. Segurització del servei

Per a accedir al servei és necessari una autenticació de tipus BASIC amb usuari/contrasenya.

També es requereix que l'usuari utilitzat per a accedir al servei tengui assignat el rol DIS_BSTWS.

2.4.3. Dependències Maven

Per a desenvolupar un client per al servei de recobriment el més senzill és crear un nou projecte amb Maven i incloent els següents blocs a dins el fitxer pom.xml:

A dins la secció <dependencies>:

```
< dependency >
    < groupId > es.caib.distribucio </ groupId >
    < artifactId > distribucio-ws-client </ artifactId >
    < version > 0.9.30 </ version >
</ dependency >
```

A dins la secció <repositories>:

```
< repository >
    < id > github-governib-maven-repo </ id >
    < name > GitHub GovernIB Maven Repository </ name >
    < url > http://GovernIB.github.io/maven/maven/ </ url >
</ repository >
```

2.5. Client d'exemple

Aquest és el codi font d'una classe de test que empra el client del servei d'enviament a bústies per a fer l'enviament d'una anotació de registre d'entrada a l'aplicació de DISTRIBUCIÓ:

```
public class BustiaV1Test {

    private static final String ENTITAT_DIST_CODI = "E00003601" ;
    private static final String UNITAT_ADM_CODI = "E03029603" ;
    private static final String APLICACIO_CODI = "CLIENT_TEST" ;
    private static final String APLICACIO_VERSIO = "2" ;
    private static final String ASSUMPTE_CODI = "A1" ;
    private static final String PROCEDIMNT_CODI = "E5" ;
    private static final String ASSUMPTE_DESC = "Descripció assumpte " ;
    private static final String ASSUMPTE_TIPUS_CODI = "A1" ;
    private static final String ASSUMPTE_TIPUS_DESC = "Assumpte de proves" ;
    private static final String USUARI_CODI = "u 123456 " ;
    private static final String USUARI_NOM = "Usuari Test " ;
    private static final String EXTRACTE = "Anotació de proves " ;
    private static final String ENTITAT_CODI = "ENT01 " ;
    private static final String ENTITAT_DESC = "Descripció entitat" ;
    private static final String OFICINA_CODI = "10" ;
    private static final String OFICINA_DESC = "Oficina de proves" ;
    private static final String LLIBRE_CODI = "10" ;
    private static final String LLIBRE_DESC = "Oficina de proves" ;
    private static final String REGISTRE_TIPUS = "E" ;
    private static final String IDIOMA_CODI = "1" ;
    private static final String IDIOMA_DESC = "Català" ;
```

```
private static final String IDENTIFICADOR = " L18E517/2018 ";
private static final String EXPEDIENT_NUM = "12345678" ;

@Test
public void test() throws DatatypeConfigurationException, IOException {
    Random generator = new Random();
    int randomNumber = generator.nextInt(9999) + 1;
    RegistreAnotacio anotacio = new RegistreAnotacio();
    anotacio.setAplicacioCodi( APLICACIO_CODI );
    anotacio.setTipusES( REGISTRE_TIPUS );
    anotacio.setAplicacioVersio( APLICACIO_VERSIO );
    anotacio.setProcedimentCodi( PROCEDIMENT_CODI );
    anotacio.setAssumppteCodi( ASSUMPTE_CODI );
    anotacio.setAssumppteDescripcio( ASSUMPTE_DESC );
    anotacio.setAssumppteTipusCodi( ASSUMPTE_TIPUS_CODI );
    anotacio.setAssumppteDescripcio( ASSUMPTE_TIPUS_DESC );
    anotacio.setUsuariCodi( USUARI_CODI );
    anotacio.setUsuariNom( USUARI_NOM );
    anotacio.setData(
        DatatypeFactory.newInstance().newXMLGregorianCalendar(
            new GregorianCalendar());
    anotacio.setExtracte( EXTRACTE );
    anotacio.setEntitatCodi( ENTITAT_CODI );
    anotacio.setEntitatDescripcio( ENTITAT_DESC );
    anotacio.setOficinaCodi( OFICINA_CODI );
    anotacio.setOficinaDescripcio( OFICINA_DESC );
    anotacio.setLlibreCodi( LLIBRE_CODI );
    anotacio.setLlibreDescripcio( LLIBRE_DESC );
    anotacio.setNumero(String.valueOf( randomNumber ));
    anotacio.setIdiomaCodi( IDIOMA_CODI );
    anotacio.setIdiomaDescripcio( IDIOMA_DESC );
    anotacio.setIdentificador( IDENTIFICADOR );
    anotacio.setExpedientNumero( EXPEDIENT_NUM );
    anotacio.setPresencial( true );
    RegistreAnnex annex1 = crearAnnex(
        "Annex1" ,
        "annex.pdf" ,
        "application/pdf" ,
        null ,
        getContingutAnnexSenseFirma(),
        "0" ,
        "EE01" ,
        "TD01" ,
        "01" );
    anotacio.getAnnexos().add( annex1 );
    RegistreAnnex justificant = crearAnnex(
        "justificant" ,
```

```

        "justificant.pdf" ,
        "application/pdf" ,
        null ,
        getContingutJustificant() ,
        "1" ,
        "EE01" ,
        "TD02" ,
        "02" );
    anotacio.setJustificant( justificant );
    try {
        getBustiaServicePort().enviarAnotacioRegistreEntrada(
            ENTITAT_DIST_CODI ,
            UNITAT_ADM_CODI ,
            anotacio );
    } catch (Exception ex) {
        ex.printStackTrace();
        fail();
    }
}

private RegistreAnnex crearAnnex(
    String titol ,
    String arxiuNom ,
    String arxiuTipusMime ,
    String arxiuUuid ,
    InputStream arxiuContingut ,
    String eniOrigen ,
    String eniEstatElaboracio ,
    String eniTipusDocumental ,
    String sicresTipusDocument ) throws IOException,
DatatypeConfigurationException {
    RegistreAnnex annex = new RegistreAnnex();
    annex.setTitol( titol );
    annex.setFitxerNom( arxiuNom );
    annex.setFitxerTipusMime( arxiuTipusMime );
    if ( arxiuContingut != null ) {
        annex.setFitxerContingut( IOUtils.toByteArray( arxiuContingut ) );
        annex.setFitxerTamany(
            annex.getFitxerContingut().length );
    }
    annex.setFitxerArxiuUuid( arxiuUuid );
    annex.setEniDataCaptura(
        DatatypeFactory.newInstance().newXMLGregorianCalendar(
            new GregorianCalendar() );
    annex.setEniOrigen( eniOrigen );
    annex.setEniEstatElaboracio( eniEstatElaboracio );
    annex.setEniTipusDocumental( eniTipusDocumental );

```



```

annex.setSicresTipusDocument( sicresTipusDocument );

MetaDades metaDades = new MetaDades();
Entry entry = new Entry();
entry.setKey( "eni:resolucion" );
entry.setValue( "12" );
metaDades.getEntry().add(entry);
entry = new Entry();
entry.setKey( "eni:profundidad_color" );
entry.setValue( "6400" );
metaDades.getEntry().add(entry);

return annex ;
}

private void afegirInteressats(RegistreAnotacio anotacio) {
    RegistreInteressat representant = new RegistreInteressat();
    representant.setAdresa( "Carrer companys" );
    representant.setCanalPreferent( "02" );
    representant.setCodiPostal( "07200" );
    representant.setDocumentNum( "77788899P" );
    representant.setDocumentTipus( "N" );
    representant.setEmail( "representant@limit.es" );
    representant.setEmailHabilitat( "true" );
    representant.setLlinatge1( "LlinatgeRep1" );
    representant.setLlinatge2( "LlinatgeRep2" );
    representant.setNom( "NomRep" );
    representant.setObservacions( null );
    representant.setPais( "Espanya" );
    representant.setPaisCodi( "ES" );
    representant.setProvincia( "Illes Balears" );
    representant.setProvinciaCodi( "022" );
    representant.setMunicipi( "Felanitx" );
    representant.setMunicipiCodi( "11" );
    representant.setRaoSocial( null );
    representant.setRepresentant( null );
    representant.setTelefon( "666555444" );
    representant.setTipus( "2" );
    RegistreInteressat interessat = new RegistreInteressat();
    interessat.setAdresa( "Carrer del moix 2" );
    interessat.setCanalPreferent( "02" );
    interessat.setCodiPostal( "07500" );
    interessat.setDocumentNum( "12312312P" );
    interessat.setDocumentTipus( "N" );
    interessat.setEmail( "interessat@limit.es" );
    interessat.setEmailHabilitat( "true" );
    interessat.setLlinatge1( "LlinatgeInt1" );

```

```

    interessat .setLlinatge2( "LlinatgeInt2" );
    interessat .setNom( "NomInt1" );
    interessat .setObservacions( null );
    interessat .setPais( "Espanya" );
    interessat .setPaisCodi( "ES" );
    interessat .setProvincia( "Illes Balears" );
    interessat .setProvinciaCodi( "11" );
    interessat .setMunicipi( "Manacor" );
    interessat .setMunicipiCodi( "033" );
    interessat .setRaoSocial( null );
    interessat .setRepresentant( representant );
    interessat .setTelefon( "999888777" );
    interessat .setTipus( "2" );
    interessat .setCodiDire( "r444444444" );
    anotacio .getInteressats().add( interessat );
  }

  private BustiaV1 getBustiaServicePort() throws IOException {
    Properties testProperties = getTestProperties();
    return BustiaV1WsClientFactory. getWsClient (
      testProperties .getProperty( "bustia.test.service.url"
    ),
      testProperties .getProperty(
"bustia.test.service.username" ),
      testProperties .getProperty(
"bustia.test.service.password" ));
  }

  private InputStream getContingutAnnexSenseFirma() {
    InputStream is = getClass().getResourceAsStream(
      "/annex_sense_firma.pdf" );
    return is ;
  }

  private InputStream getContingutJustificant() {
    InputStream is = getClass().getResourceAsStream(
      "/justificant.pdf" );
    return is ;
  }

  private Properties getTestProperties() throws IOException {
    Properties props = new Properties();
    InputStream is = getClass().getResourceAsStream(
      "/bustia_test.properties" );
    props .load( is );
    return props ;
  }
}

```

Aquesta classe llegeix la configuració per a la connexió amb el servei del fitxer bustia_test.properties. El contingut d'aquest fitxer és el següent:

```
bustia.test.service.url= http://localhost:8080/distribucio/ws/v1/bustia  
bustia.test.service.username= usuari  
bustia.test.service.password= contrasenya
```

3. Backoffices de DISTRIBUCIO

Les aplicacions que s'integren com a backoffices de Distribució implementen una API REST o WS SOAP per rebre comunicacions d'anotacions pendents (3.1) i consumeixen l'API REST o WS SOAP de Distribució per a consultar el detall de les anotacions (3.2) i canviar l'estat a les anotacions conforme les marca com a rebudes, rebutjades o processades (3.3). Per entendre el procés es pot consultar el punt 3.4 Diagrama de funcionament .

3.1. Distribució d'anotacions cap a backoffices

Mitjançant regles és possible distribuir automàticament anotacions de registre que arriben a DISTRIBUCIÓ cap a aplicacions de backoffice de DISTRIBUCIO.

Per a poder distribuir cap a una aplicació aquesta ha d'implementar el WS de distribució per rebre notificacions de noves anotacions de registre i emprar el WS d'integració de DISTRIBUCIO per recuperar la informació de les anotacions de registre notificades.

A partir de la versió 0.9.44 la consulta d'anotacions i canvis d'estat per part dels backoffices es pot fer invocant la API REST de DISTRIBUCIO i a partir de la versió 0.9.45 la comunicació d'anotacions pendents es pot rebre des de DISTRIBUCIO exposant una API REST en els backoffices. Per compatibilitat amb els backoffices existents els serveis web SOAP de tramitació i de comunicació d'anotacions continuen estant actius però són obsolets i es recomana l'ús de comunicació per API REST.

Per la distribució automàtica de registres s'han de crear regles de tipus backoffice on tenim els següents camps:

- **Nom:** És el nom de la regla i es tracta d'un camp obligatori.
- **Descripció:** Camp opcional on es pot introduir informació de la regla.
- **Unitat organitzativa:** Aquest camp és opcional i s'emplea com a filtre, fa referència a la unitat organitzativa, en cas de tenir un valor la regla l'haurà de tenir en compte per poder executar-se.
- **Codis SIA:** Aquest camp és obligatori i el seu valor a de ser únic respecte a les altres regles. Es poden afegir tots els codis SIA que es vulguin sempre i quan siguin diferents i únics.
- **Tipus:** Camp obligatori on s'ha de posar el tipus de regla, en aquest cas 'Gestionar amb backoffice'.
- **Backoffice:** Camp obligatori on es posarà el codi backoffice on es vol enviar l'anotació.

Les regles actives de tipus backoffice s'executaran quan Distribució rebí una anotació amb un codi de procediment que sigui igual al codi SIA de la regla, a més de l'unitat organitzativa en cas de que la regla tingui el camp omplert, i l'enviarà al backoffice que marqui la regla.

Per facilitar la creació de regles existeix l'API REST de regles que permet crear, consultar i activar/desactivar les regles. Aquesta API es detalla en el punt 3.1.4.

3.1.1. Integració com a backoffice de DISTRIBUCIO

Els backoffices de DISTRIBUCIO hauran d'implementar un servei que permeti avisar de l'arribada d'anotacions de registre pendents de processar. La interfície d'aquest servei serà la següent:

<<Interface>> DistribucioBackoffice	AnotacioRegistreId
+ comunicarAnotacionsPendents(List<AnotacioRegistreId> ids): void	+ identificador: String + clauAcces: String

A l'Annex II: WSDL del backoffice es descriu el WSDL necessari per a la seva implementació.

Pels backoffices que estiguin configurats com a de tipus API REST hauran d'exposar un mètode per rebre el JSON de la llista d'identificadors i l'autenticació pot ser WSBasic amb autenticació bàsica.

L'exemple de JSON que s'enviarà en el cos de la petició POST és la següent:

taula. Exemple de JSON que es comunicarà als backoffices de tipus API REST per comunicar anotacions pendents.

```
[ {
  "identificador" : "GOIBE1111/2022",
  "clauAcces" : "hv/8hXYRSFYtVEjZyDfEE8Kd0IiX3N1UEXGilt13rbw="
}, {
  "identificador" : "GOIBE2222/2022",
  "clauAcces" : "AzRwN0QQV70RR4IxXFZsPj2FYJ2VA7LDiKap19MmA/U="
} ]
```

3.1.2. WS d'integració amb els backoffices

L'aplicació DISTRIBUCIO implementa un servei que permet als backoffices obtenir informació d'una anotació de registre i canviar el seu estat. La interfície d'aquest servei és la següent:

<<Interface>> BackofficeIntegracio	
+ consulta(AnotacioRegistreId id): AnotacioRegistreEntrada	
+ canviEstat(AnotacioRegistreId id, Estat estat): void	

AnotacioRegistreId
+ identificador: String
+ clauAcces: String

<<Enum>> Estat
+ PENDENT
+ PROCESSADA
+ ERROR

En el punt 3.5 es detalla el model de dades per a la seva invocació i a l' Annex III: WSDL del servei d'integració s'hi troba el contingut del WSDL per a la implementació del client del WS.

A continuació explicam amb més detall els camps de cada una de les classes del model de dades pel mètode de consulta.

annex				
Nom del camp	Tipus	Mult.	Tamany	Descripció
contingut	Byte[]	0..1	var	Cadena de bytes que componen el contingut del fitxer de firma annex si n'hi ha. És obligatori si el tipus de firma és TF04.
firmaContingut	Byte[]	0..1	var	Cadena de bytes que componen el contingut del fitxer de firma annex si n'hi ha. És obligatori si el tipus de firma és TF04.
firmaNom	Cadena	0..1	256	Nom del fitxer de la firma de l'annex si n'hi ha. És obligatori si el tipus de firma és TF04.
firmaPerfil	firmaPerfil	0..1	30	Codi de perfil de la firma.
firmaTamany	Numèric			
firmaTipus	firmaTipus	0..1	30	Codi del tipus de firma.
firmaTipusMime	Cadena	0..1	30	Tipus MIME del fitxer de firma annexat si n'hi ha. És obligatori si el tipus de firma és TF04.
nom	Cadena	0..1	256	Nom del fitxer annexat.
ntiEstadoElaboracion	ntiEstadoElaboracion	0..1	4	Codi de la metadata estat elaboració de la NTI de document electrònic.
ntiFechaCaptura	dateTime	0..1	14	Data de captura del document.
ntiOrigen	ntiOrigen	0..1	10	Codi de la metadata origen de la NTI de document electrònic.
ntiTipoDocumental	ntiTipoDocumento	0..1	255	Codi de la metadata tipus documental de la NTI de document

				electrònic.
observacions	Cadena	0..1	50	Observacions a destacar sobre la informació annexada.
sicresTipoDocumento	sicresTipoDocumento	0..1	2	Codi SICRES del segment Anexo – tipo de documento.
sicresValidezDocumento	sicresValidezDocumento	0..1		
tamany	numèric			Tamany en bytes del fitxer annexat.
tipusMime	Cadena	0..1	30	Tipus MIME del fitxer annexat.
títol	Cadena	0..1	200	Títol del document.
uuid	Cadena	0..1	256	Uuid de l'arxiu digital si s'escau.

anotacioRegistreBase				
Nom del camp	Tipus	Mult.	Tamany	Descripció
annexos	annex	0..*		Annex al que fa referència amb les seves propietats.
aplicacioCodi	Cadena	0..1	255	Codi de l'aplicació que fa l'enviament de l'assentament registral.
aplicacioVersio	Cadena	0..1	255	Versió de l'aplicació que fa l'enviament de l'assentament registral.
assumpteCodiCodi	Cadena	0..1	16	Codi de l'assumpte.
assumpteCodiDescripcio	Cadena	0..1	255	Denominació de l'assumpte.
assumpteTipusCodi	Cadena	0..1	16	Codi de l'assumpte.
assumpteTipusDescripcio	Cadena	0..1	100	Denominació del tipus d'assumpte de l'assentament registral.
data	dateTime	0..1	14	Data de l'assentament.
docFisicaCodi	Cadena	0..1	19	Codi SICRES del segment internos – Documentación física y/o soportes.
docFisicaDescripcio	Cadena	0..1	100	Descripció del segment internos – Documentación física y/o soportes.
entitatCodi	Cadena	0..1	255	Codi DIR3 de l'entitat governamental a la que va dirigit l'assentament registral.
entitatDescripcio	Cadena	0..1	255	Descripció de l'entitat governamental a la que va dirigit l'assentament registral.
expedientNumero	Cadena	0..1	255	Número identificador de l'assentament registral. És únic.
exposa	Cadena	0..1	lob	Exposició de fets.
extracte	Cadena	0..1	240	Extracte o resum de l'assentament.

identificador	Cadena	0..1	100	Codi identificador de l'assentament registral.
idiomaCodi	Cadena	0..1	2	Codi d'idioma amb el que s'ha fet l'assentament.
idiomaDescripcio	Cadena	0..1	100	Descripció d'idioma amb el que s'ha fet l'assentament.
interessats	interessat	0..*		Dades que componen els interessats de l'assentament registral si n'hi ha.
llibreCodi	Cadena	0..1	4	Codi del llibre al qual s'ha fet l'assentament registral rebut.
llibreDescripcio	Cadena	0..1	255	Denominació del llibre al qual s'ha fet l'assentament registral rebut.
observacions	Cadena	0..1	50	Observacions de l'assentament.
oficinaCodi	Cadena	0..1	21	Codi DIR3 de l'oficina on es registre.
oficinaDescripcio	Cadena	0..1	300	Denominació DIR3 de l'entitat registral on es registre l'assentament.
origenData	dateTime	0..1	14	Data origen de l'assentament.
origenRegistreNumero	Cadena	0..1	80	
procedimentCodi	Cadena	0..1	64	Codi de procediment.
refExterna	Cadena	0..1		
solicita	Cadena	0..1	lob	Objecte de la sol·licitud si escau.
transportNumero	Cadena	0..1	20	Número de transport.
transportTipusCodi	Cadena	0..1	20	Codi SICRES3 del segment internos – tipo transporte.
transportTipusDescripcio	Cadena	0..1	100	Descripció del segment internos – tipo transporte.
usuariCodi	Cadena	0..1	20	Codi de l'usuari que vol realitzar l'operació.
usuariNom	Cadena	0..1	767	Nom de l'usuari que vol realitzar l'operació.
justificantFitxerArxiuUuid	Cadena	0..1	256	Paràmetre que emmagatzema el uuid del justificant. Serveix per recuperar el justificant des del backoffice.

AnotacioRegistreEntrada => tns:anotacioRegistreBase

Nom del camp	Tipus	Mult.	Tamany	Descripció
destiCodi	Cadena	0..1	21	Codi DIR3 de la oficina destí.
destiDescripcio	Cadena	0..1	300	Descripció de la oficina destí.

anotacioRegistreId				
Nom del camp	Tipus	Mult.	Tamany	Descripció
clauAcces	Cadena	0..1		Clau d'accés per accedir a l'informació.
identificador	Cadena	0..1		Identificador de l'usuari per poder accedir a l'informació.

interessatBase				
Nom del camp	Tipus	Mult.	Tamany	Descripció
adresa	Cadena	0..1	160	Adreça postal de l'interessat.
adresaElectronica	Cadena	0..1	160	Adreça electrònica de l'interessat.
canal	Cadena	0..1	2	Codi SICRES del segment interesado – canal preferente de notificació del interesado.
cp	Cadena	0..1	5	Codi postal del l'interessat.
documentNumero	Cadena	0..1	17	Número de document identificatiu.
documentTipus	documentTipus	0..1	1	Codi SICRES del segment interesado – documento de identificacion de interesado tipo.
email	Cadena	0..1	160	Correu electrònic de l'interessat.
l·linatge1	Cadena	0..1	255	Primer l·linatge de l'interessat. Obligatori si és una persona física.
l·linatge2	Cadena	0..1	255	Segon l·linatge de l'interessat.
municipi	Cadena	0..1	100	Nom del municipi de la persona.
municipiCodi	Cadena	0..1	100	Codi del municipi de la persona.
nom	Cadena	0..1	255	Nom de l'interessat. Obligatori si és una persona física.
observacions	Cadena	0..1	160	Observacions a destacar sobre la persona interessada.
organCodi	Cadena	0..1		
pais	Cadena	0..1	100	Nom del País de la persona.
paisCodi	Cadena	0..1	100	Codi ISO numèric del País de la persona.
provincia	Cadena	0..1	100	Nom de la província de la persona.
provinciaCodi	Cadena	0..1	100	Codi INE de la província de la

				persona.
raoSocial	Cadena	0..1	2000	Raó social de l'interessat. Obligatori si és una persona jurídica.
telefon	Cadena	0..1	20	Telèfon de l'interessat.
tipus	interessatTipus	0..1	19	Figura jurídica de la persona '1' = Administració. '2' = Persona física. '3' = Persona jurídica.

Interessat => tns:interessatBase				
Nom del camp	Tipus	Mult.	Tamany	Descripció
representant	representant	0..1		Dades de la persona representant descrites amb les mateixes propietats que els interessats.

3.1.3. API REST d'integració amb backoffice

L'API REST d'integració amb els backoffices és l'adaptació del WS SOAP d'integració amb els backoffices del punt anterior 3.1.2 per poder ampliar la informació dels annexos i poder afegir informació sobre si els annexos són vàlids i si el seu estat a l'Arxiu és esborrany o definitiu.

Així doncs, aquesta API REST permet als backoffices integrar-se amb DISTRIBUCIO per poder consultar les anotacions comunicades per part de DISTRIBUCIO i canviar-ne l'estat depenent de si s'han pogut rebre les anotacions, processar-les o rebutjar-les.

Les dades bàsiques d'aquesta API REST d'integració amb els backoffices són les següents:

- URL base: </distribucioapi/interna/backoffice>
- Documentació Swagger: <https://se.caib.es/distribucioapi/interna/api/rest>
- Autenticació: WS_BASIC
- Rol necessari: DIS_BACKWS

A continuació es descriuen els mètodes de l'API REST per consultar i canviar l'estat i posteriorment es presenta la llibreria del client REST que es pot incloure com a dependència *maven* i un exemple del seu ús.

Consulta d'una anotació

Aquest mètode és el que permet consultar el detall d'una anotació de registre a partir de les dades comunicades per DISTRIBUCIO formades per l'identificador de l'anotació que sol ser el número de registre i una clau codificada per restringir-ne la consulta. Segons la configuració de DISTRIBUCIO els annexos poden tenir contingut o no però típicament es retorna la informació dels annexos sense contingut per a que sigui el propi backoffice que consulti l'annex a l'Arxiu a partir del seu identificador UUID.

URL:	/backoffice/consulta
Tipus:	GET
Descripció:	Consulta les dades d'una anotació i dels seus interessats i annexos a partir de l'identificador i clau comunicada per DISTRIBUCIO.
Paràmetres:	<ul style="list-style-type: none"> identificador: És l'identificador de l'anotació comunicada per DISTRIBUCIO. Sol coincidir amb el número de registre de l'anotació. clauAcces: És una clau proporcionada per DISTRIBUCIO en el moment de notificar al backoffice l'anotació de registre que s'ha d'emprar per consultar o canviar l'estat de l'anotació. <p>Com que és una petició de consulta els paràmetres s'informen en la pròpia URL. Un exemple de petició seria el següent. Pels paràmetres identificador: "GOIBE715-DEV/2023" i clau d'accés "JRwLmCaMES6JKAAnzhd5K9Qi4ppjudiLwd+2b5LNsVo=" la petició amb els paràmetres codificats és:</p> <p>GET</p> <p>https://se.caib.es/distribucioapi/interna/backoffice/consulta?identificador=GOIBE715-DEV%2F2023&clauAcces=JRwLmCaMES6JKAAnzhd5K9Qi4ppjudiLwd%2B2b5LNsVo%3D</p>
Resultat:	<ul style="list-style-type: none"> HTTP 200 i el JSON del detall de l'anotació com a cos de la resposta. HTTP 401 si l'usuari no té el rol DIS_BACKWS amb un missatge d'error com a cos de la resposta. HTTP 500 si es produeix un error intern no controlat consultant l'anotació. <p>Un exemple de JSON de resposta, de forma abreujada, seria el següent:</p> <pre>{ "identificador": "GOIBE715-DEV/2023", "procedimentCodi": "703381", "extracte": "Solicitud de equivalencia - Equivalencia con los títulos de Graduado en Educación Secundaria Obligatoria y de Bachiller", "data": 1683181017253, ... "interessats": [{ "tipus": "PERSONA_FISICA", "documentTipus": "NIF",</pre>

```

...
  },
],
"annexos": [
  {
    "titol": "Formulario de identificación del firmante",
    "nom": "FORM-1-1.pdf",
    "contingut": null,
    "uuid": "0276698a-d0f9-44f7-84be-841a98881e4a",
    ...
    "documentValid": true,
    "documentError": null,
    "estat": "DEFINITIU"
  }
],
"justificantFitxerArxiuUuid": "0cd70aa5-77aa-4d2e-8032-d84cad38669b",
"destiCodi": "A04013522",
"destiDescripcio": "Direcció General de Planificació, Ordenació i Centres"
}

```

El JSON de la resposta conté les dades de l'anotació, els interessats i dels annexos. És el mapeig a JSON de les classes descrites en el punt anterior del WS d'integració.

Canvi d'estat d'una anotació

Aquest mètode és el que han d'invocar els backoffices de DISTRIBUCIO per informar dins de DISTRIBUCIO de l'estat en què es troben les anotacions quan es rebin, processin amb o sense error o es rebutgin. De forma semblant a la consulta s'haurà d'usar l'identificador de l'anotació amb la clau d'accés.

URL:	/backoffice/canviEstat
Tipus:	POST
Descripció:	Canvia l'estat d'una anotació de registre enviada a backoffice.
Paràmetres:	<p>En aquest cas la petició es passa com a un objecte JSON en el cos de la petició POST on s'ha d'especificar l'identificador amb la clau, l'estat i les observacions. Els paràmetres serien els següents:</p> <ul style="list-style-type: none"> • id: <ul style="list-style-type: none"> ◦ identificador: És l'identificador de l'anotació comunicada per DISTRIBUCIO. Típicament sol ser el número de registre de l'anotació. Format String, dada obligatòria. ◦ clauAcces: És una clau proporcionada per DISTRIBUCIO en el moment

	<p>de notificar al backoffice l'anotació de registre que s'ha d'emprar per consultar o canviar l'estat a l'anotació. Format String, dada obligatòria.</p> <ul style="list-style-type: none"> • estat: Codi del nou estat a comunicar a DISTRIBUCIO per l'anotació. S'ha d'informar en format String i és una dada obligatòria. Els possibles estats són: <ul style="list-style-type: none"> ◦ REBUDA: Indica a DISTRIBUCIO que pot marcar l'anotació com a rebuda al backoffice. ◦ PROCESSADA: Marca l'anotació l'anotació coma processada en el backoffice i per tant és un estat final. DISTRIBUCIO programarà el tancament de l'expedient de l'Arxiu digital per aquella anotació, per la qual cosa és convenient haver mogut tots els annexos que es vulguin incloure a l'expedient de l'Arxiu de l'expedient del backoffice. ◦ REBUTJADA: Marca l'anotació com a rebutjada pel backoffice. Aquesta apareixerà com a rebutjada pel backoffice a DISTRIBUCIO amb el comentari que s'hagi informat. ◦ ERROR: Marca l'anotació com a processada amb error al BACKOFFICE. Aquestes anotacions apareixeran en el llistat d'anotacions pendents i pot ser que es reenviïn cap al backoffice per reintentar el processament. • observacions: És una cadena de text opcional en format String que es pot enviar juntament amb el canvi d'estat. En el cas de les anotacions rebutjades o processades amb error es pot visualitzar aquest comentari a DISTRIBUCIO així que és recomanable informar del resultat de l'operació que acompanya al canvi d'estat. La grandària màxima són 4000 caràcters. <p>Un exemple de petició seria el següent. Pels paràmetres identificador: "GOIBE1682590959780/2023" i clau d'accés "RfG3stK7Bc7JmuOk+riod/uOnBNGUEPqZt6+1/HiTlc=" la petició amb els paràmetres en el cos de la petició POST com a JSON seria:</p> <pre>POST { "id": { "identificador": "GOIBE1682590959780/2023", "clauAcces": "RfG3stK7Bc7JmuOk+riod/uOnBNGUEPqZt6+1/HiTlc=" }, "estat": "PROCESSADA", "observacions": "S'ha creat l'expedient 123/2023" }</pre>
Resultat:	<ul style="list-style-type: none"> • HTTP 200 si la petició s'ha processat correctament. • HTTP 500 si es produeix un error intern no controlat consultant l'anotació. Com a cos de la resposta es retorna el missatge de confirmació.

El JSON de la resposta conté les dades de l'anotació, els interessats i dels annexos. És el mapeig a JSON de les classes descrites en el punt anterior del WS d'integració.

Per facilitar l'ús de l'API REST d'integració per backoffices s'ofereix un client REST que permet consultar i canviar l'estat fent ús de classes Java amb un client Jersey Client versió 1.8 que manté l'autenticació WS BASIC i suporta redireccions. És possible incloure aquest client amb la següent dependència maven del repositori CAIB.

taula. Dependència maven del client REST de l'API REST per backoffices de consulta i canvi d'estat d'anotacions.

```
<dependency>
  <groupId>es.caib.distribucio</groupId>
  <artifactId>distribucio-rest-client</artifactId>
  <version>0.9.47</version>
</dependency>
```

A continuació es mostra un exemple Java per configurar el client REST, consultar una anotació i canviar l'estat a rebuda.

taula. Exemple d'ús del client REST per integració de backoffices per configurar, consultar i canviar l'estat d'una anotació.

```
package es.caib.distribucio.rest.client;

import es.caib.distribucio.rest.client.domini.AnotacioRegistreEntrada;
import es.caib.distribucio.rest.client.domini.AnotacioRegistreId;
import es.caib.distribucio.rest.client.domini.Estat;

/** Test per provar el client REST de l'API REST d'integració per backoffices. Es
    prova la consulta i canvi d'estat d'una anotació.
    */
public class ApiRestIntegracioTest {

    public static String BASE_URL = "https://se.caib.es/distribucioapi/interna";
    public static String USERNAME = "$helium_distribucio";
    public static String PASSWORD = "*****";

    /** Prova de crear una regla, descactivar-la i consultar-la.
        * @throws Exception */
    public static void main(String[] args) throws Exception {

        // Configura el client API REST
        BackofficeIntegracioRestClient restClient =

        BackofficeIntegracioRestClientFactory.getRestClient(BASE_URL, USERNAME,
        PASSWORD);

        // Crea l'objecte id per la consulta
        AnotacioRegistreId id = new AnotacioRegistreId();
        id.setIndetificador("GOIBE715-DEV/2023");
```

```
id.setClauAcces("JRwLmCaMES6JKAAnzhd5K9Qi4ppjudiLwd+2b5LNsVo=");

// Consulta l'anotació
AnotacioRegistreEntrada anotacio = restClient.consulta(id);
System.out.println("Anotació consultada: " + anotacio.getIdentificador()
+ " " + anotacio.getExtracte());

// Canvi d'estat
restClient.canviEstat(id, Estat.REBUDA, "Observacions text");
System.out.println("Test finalitzat correctament.");
}
```

En la següent taula es mostra la sortida per consola del test anterior.

taula. Sortida per consulta de la consulta i canvi d'estat.

```
Anotació consultada: G0IBE715-DEV/2023 Solicitud de equivalencia - Equivalencia con
los títulos de Graduado en Educación Secundaria Obligatoria y de Bachiller
Test finalitzat correctament.
```

3.1.4. API REST de creació de regles

Per tal de facilitar la creació automàtica de regles de tipus enviar a backoffice segons un codi SIA de procediment DISTRIBUCIO ofereix una API REST de creació, consulta i modificació de regles. Les dades bàsiques d'aquesta API REST de regles són les següents:

- URL base: <https://se.caib.es/distribucioapi/interna/regla>
- Documentació Swagger: <https://se.caib.es/distribucioapi/interna/api/rest>
- Autenticació: WSBASIC
- Rol necessari: DIS_REGLA

Les operacions que es poden realitzar són les següents:

Creació de regles

Aquest mètode és el que permet donar d'alta una regla per un backoffice i un codi de procediment SIA en una entitat de DISTRIBUCIO.

URL:	/regla/add
Tipus:	POST
Descripció:	Dona d'alta una regla pel backoffice i codi SIA indicat per a l'entitat indicada.
Paràmetres:	<ul style="list-style-type: none"> • entitat: Codi de l'entitat de DISTRIBUCIO on crear la regla. Típicament per l'entorn del Govern el codi d'entitat és "A04003003". Aquest paràmetre és obligatori. • sia: Codi de procediment SIA per crear la regla i aplicar-la a les anotacions

	<p>que es donin d'alta amb aquest codi de procediment. Aquest paràmetre és obligatori.</p> <ul style="list-style-type: none"> • backoffice: Codi del backoffice pel qual donar d'alta la regla. El backoffice ha d'estar donat d'alta en el manteniment de backoffices a DISTRIBUCIO per l'entitat indicada. Aquest paràmetre és obligatori.
Resultat:	<ul style="list-style-type: none"> • HTTP 200 si la regla s'ha creat correctament o ja existia pel mateix backoffice i com a cos de la resposta es retorna el text de missatge de confirmació. • HTTP 404 si l'entitat o el backoffice no es troba per codi. • HTTP 406 si ja existeix una regla pel mateix codi de procediment però per un altre backoffice. • HTTP 500 si es produeix un error intern no controlat creant la regla.

Canvi d'estat

Aquest mètode permet activar o desactivar una regla de tipus enviament a backoffice a partir del codi SIA del procediment.

URL:	/regla/canviEstat
Tipus:	POST
Descripció:	Activa o desactiva una regla a partir del codi SIA de procediment.
Paràmetres:	<ul style="list-style-type: none"> • sia: Codi de procediment SIA per trobar la regla associada. Aquest paràmetre és obligatori. • activa: Paràmetre opcional de tipus booleà per activar o desactivar la regla. Si on s'especifica es canvia segons el valor que tingui actualment. Per exemple <i>activa=true</i>.
Resultat:	<ul style="list-style-type: none"> • HTTP 200 si la regla s'ha pogut modificar correctament. Com a cos del missatge de resposta es retorna el text de confirmació conforme si la regla s'ha activat o desactivat correctament. Per exemple "La regla amb codi 12354 s'ha desactivat correctament." • HTTP 409 si no s'ha trobat cap regla el codi de procediment SIA. • HTTP 500 si es produeix un error intern no controlat modificant la regla.

Consultar regla

Aquest mètode permet consultar els detalls d'una regla a partir del codi SIA del procediment. En el cos de la resposta es retorna un JSON amb el detall de la regla.

URL:	/regla/consultarRegla
Tipus:	GET
Descripció:	Consulta les regles per codi SIA que existeixin. En principi només hi pot haver una regla de tipus backoffice per codi SIA però per compatibilitat es retorna una llista de regles que podrien estar creades pel mateix codi SIA de procediment anteriorment.
Paràmetres:	<ul style="list-style-type: none"> sia: Codi SIA de la regla que identifica la regla de tipus backoffice.
Resultat:	<ul style="list-style-type: none"> HTTP 200 si la regla s'ha pogut consultar correctament. En el cos del missatge hi ha el JSON amb la llista de regles i els seus detalls, per exemple: <ul style="list-style-type: none"> Petició: <ul style="list-style-type: none"> GET <code>https://se.caib.es/distribucioapi/interna/regla/consultarRegla?sia=20220429</code> Cos de la resposta <pre>[{ "id": 1485186, "entitat": "Govern de les Illes Balears", "data": "13/05/2022 16:51:55", "activa": true, "nom": "helium 20220429", "backofficeDesti": "HELIUM DEV" }]</pre> Els camps de la resposta són els següents: <ul style="list-style-type: none"> id: identificador intern de la regla de tipus long. entitat: Descripció de l'entitat de DISTRIBUCIO a la qual pertany la regla. Tipus String. data: Data de creació de la regla. String amb format "dd/MM/yyyy HH:mm:ss". nom: Nom posat a la regla. backofficeDesti: Codi del backoffice al qual es notifiquen les anotacions afectades per la regla. HTTP 404 si no s'ha trobat cap regla el codi de procediment SIA.

A continuació es mostra un exemple complet de crida a la API REST de regles amb un client de Jersey. Si es vol provar es pot incloure la següent dependència maven.

taula. Dependència maven per usar el client jersey de l'exemple Java de l'ús de l'API REST de regles.

```
<dependency>
  <groupId>com.sun.jersey</groupId>
  <artifactId>jersey-client</artifactId>
  <version>1.19</version>
  <scope>test</scope>
</dependency>
```

A l'exemple següent es configura el client de Jersey amb autenticació bàsica i un filtre per recuperar i reusar les cookies per evitar fer login en cada petició. A continuació crea una nova regla amb un *timestamp* com a codi SIA del procediment, la desactiva i la consulta.

taula. Exemple Java amb Jersey Client per donar d'alta una regla a DISTRIBUCIO fent ús de l'API REST de regles.

```
package es.caib.distribucio.api.interna.controller;

import java.util.ArrayList;
import java.util.Date;

import com.sun.jersey.api.client.Client;
import com.sun.jersey.api.client.ClientHandlerException;
import com.sun.jersey.api.client.ClientRequest;
import com.sun.jersey.api.client.ClientResponse;
import com.sun.jersey.api.client.filter.ClientFilter;
import com.sun.jersey.api.client.filter.HTTPBasicAuthFilter;

/** Test per provar l'API REST de regles de l'api interna usant un client Jersey. */
public class ReglaRestTest {

    public static String BASE_URL =
"https://se.caib.es/distribucioapi/interna/regla";
    public static String ENTITAT = "A04003003";
    public static String BACKOFFICE = "helium";
    public static String USERNAME = "$helium_distribucio";
    public static String PASSWORD = "*****";

    /** Prova de crear una regla, desactivar-la i consultar-la. */
    public static void main(String[] args) {

        // Configura el client jersey.
        ClientResponse response;
        Client jerseyClient = Client.create();
        // Afegeix l'autenticació WS Basic
        jerseyClient.addFilter(
            new HTTPBasicAuthFilter(USERNAME, PASSWORD));
        // Afegeix un filtre per evitar fer login en cada petició
        jerseyClient.addFilter(
            new ClientFilter() {
                private ArrayList<Object> cookies;
                @Override
                public ClientResponse handle(ClientRequest request)
```

```
throws ClientHandlerException {
    cookies);
    getNext().handle(request);
    if (cookies != null) {
        request.getHeaders().put("Cookie",
    }
    ClientResponse response =
    if (response.getCookies() != null) {
        if (cookies == null) {
            cookies = new ArrayList<Object>();
        }
        cookies.addAll(response.getCookies());
    }
    return response;
}

);

String sia = String.valueOf(new Date().getTime());

// Crea una nova regla
response = jerseyClient
    .resource(BASE_URL + "/add?entitat=" + ENTITAT + "&sia=" +
sia + "&backoffice=" + BACKOFFICE)
    .post(ClientResponse.class);
System.out.println("Resposta de la creació de la regla: " +
response.getStatus() + " " + response.getEntity(String.class));

// Desactiva la regla
response = jerseyClient
    .resource(BASE_URL + "/canviEstat?&sia=" + sia +
"&activa=false")
    .post(ClientResponse.class);
System.out.println("Resposta de la desactivació de la regla: " +
response.getStatus() + " " + response.getEntity(String.class));

// Consulta la regla
response = jerseyClient
    .resource(BASE_URL + "/consultarRegla?sia=" + sia)
    .get(ClientResponse.class);
System.out.println("Resposta de la consulta de la regla: " +
response.getStatus() + " " + response.getEntity(String.class));
}
```

La resposta per consola de l'execució del test anterior hauria de ser semblant a la següent:

taula. Sortida per consola de l'exemple anterior.

```
Resposta de la creació de la regla: 200 Regla amb id 4049906 "helium 1683294623267"
creada correctament pel backoffice helium pel codi SIA 1683294623267 a l'entitat
A04003003
Resposta de la desactivació de la regla: 200 La regla amb codi 1683294623267 s'ha
desactivat correctament.
Resposta de la consulta de la regla: 200 [{"id":4049906,"entitat":"Govern de les Illes
```

```
Balears", "data": "05/05/2023 15:49:43", "activa": false, "nom": "helium  
1683294623267", "backofficeDesti": "HELIUM DEV"}]
```

A DISTRIBUCIO es pot comprovar des del perfil d'administrador per l'entitat com la regla s'ha creat correctament.

fig. Regla creada a Distribucio.

	Nom	Codi assumpte	Codis SIA	Unitat Organitzativa	Bústia	Presencial	Destinació	Activa	
...	helium 1683294623267		1683294623267				HELIUM DEV BACKOFFICE		Accions

3.2. Mètode Consulta

Aquest mètode necessitarà un únic paràmetre de tipus anotacioRegistreId que es crea amb dos propietats:

- **anotacioRegistreId** – Es tracta d'un identificador que porta les següents propietats:
 - clauAcces, de tipus cadena (String). Es tracta de la clau (contrasenya) amb la que està registrat l'usuari.
 - Identificador, de tipus cadena (String). Es tracta de l'identificador amb el que està registrat l'usuari.

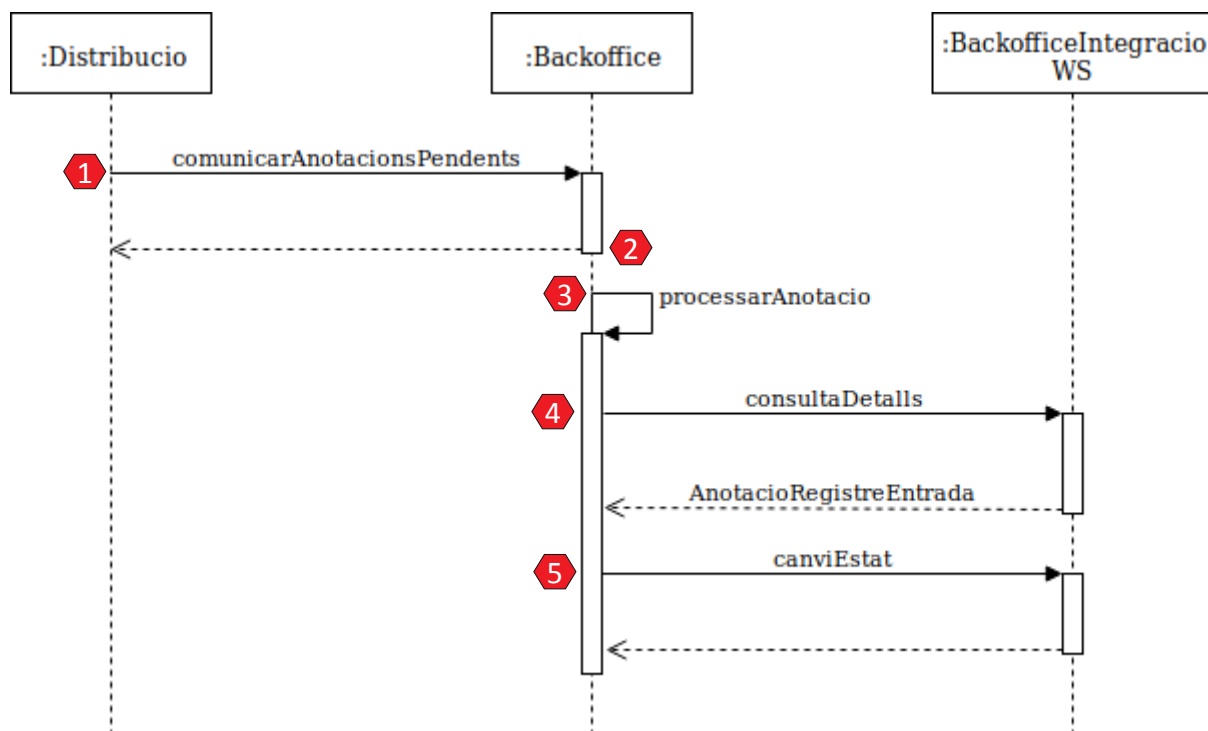
3.3. Mètode CanviEstat

Aquest mètode necessitarà tres paràmetres:

- **id** – Aquest identificador es crea amb un objecte de tipus anotacioRegistreId, que té dos propietats descrites al punt anterior (3.2).
- **estat** – Es tracta d'un paràmetre de tipus cadena (String) on s'indicarà el nou estat que volem per l'anotació.
- **observacions** – Es tracta d'un paràmetre de tipus cadena (String) on es descriuran les observacions que es volen donar per fer el canvi.

3.4. Diagrama de funcionament

El detall de la comunicació entre l'aplicació DISTRIBUCIO e el backoffice es pot veure en el següent diagrama.



1. L'aplicació de distribució comunica al backoffice l'existència d'anotacions de registre pendents de processar mitjançant el mètode comunicarAnotacionsPendants del servei DistribucioBackoffice.
2. El backoffice ha de guardar en una coa la llista d'anotacions a processar i finalitzar la comunicació.
3. El backoffice ha de processar les anotacions pendents que se li han comunicat.

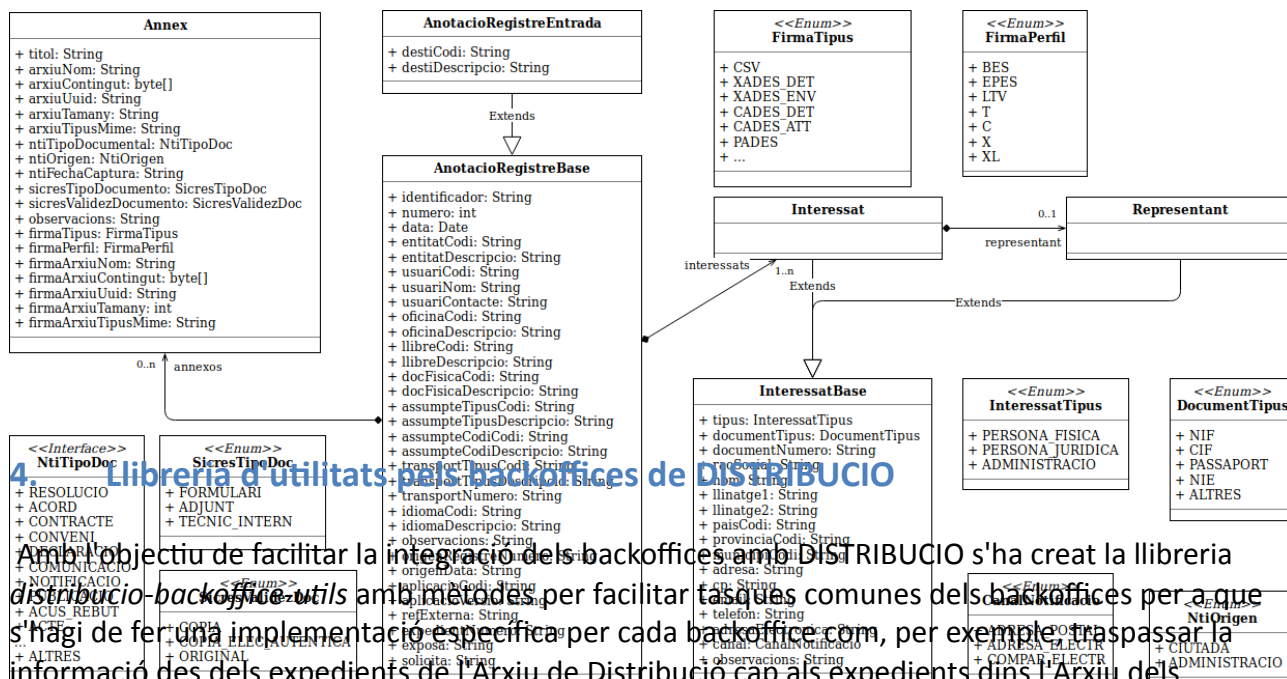
A mesura que el backoffice vagi processant les anotacions de registre, es requereix la següent comunicació amb l'aplicació de DISTRIBUCIO:

4. El backoffice consultarà la informació de l'anotació de registre al backoffice mitjançant el mètode consulta del servei BackofficeIntegracio. Com a resultat d'aquesta cridada, el backoffice obtindrà els detalls de l'anotació de registre.
5. Una vegada processada l'anotació, el backoffice haurà de comunicar a l'aplicació de DISTRIBUCIO l'estat d'aquest processament mitjançant el mètode canviEstat del servei DistribucioBackoffice. Els possibles estats son:
 - PENDENT
 - PROCESSADA
 - ERROR

Si el backoffice ha processat correctament l'anotació de registre l'estat indicat haurà de ser «PROCESSADA». En canvi, si l'anotació ha produït algun tipus d'error de processament, l'estat indicat haurà de ser «ERROR».

3.5. Model de dades

A continuació es mostra el diagrama de classes amb els detalls de l'objecte AnotacioRegistreEntrada:



4. Llibreria d'utilitats dels backoffices de DISTRIBUCIO

Amb l'objectiu de facilitar la integració dels backoffices amb DISTRIBUCIO s'ha creat la llibreria *distribucio-backoffice-utils* amb mètodes per facilitar tasques comunes dels backoffices per a que s'hagi de fer una implementació específica per cada backoffice com, per exemple, traspasar la informació des dels expedients de l'Arxiu de Distribució cap als expedients dins l'Arxiu dels backoffices o extreure dades de documents tècnics de Sistra2.

La motivació d'aquesta llibreria és unificar la forma en què els backoffices mouen la informació des dels expedients de DISTRIBUCIO cap als seus propis expedients dins l'Arxiu. D'aquesta forma les crides per moure el contingut i els interessats s'encapsula dins d'una funció que ja té en compte els diferents errors que es poden produir o si els annexos o els interessats ja s'han traspassat en un intent anterior.

En els següents punts des descriu la llibreria, com funciona i com utilitzar-la.

4.1. Informació de la llibreria

La llibreria consisteix en classes interfícies amb els mètodes de suport pels *backoffices* i classes amb la implementació. La llista de classes d'utilitats són les següents:

- *BackofficeArxiuUtils*: Aquesta classe d'utilitats conté mètodes per tractar les anotacions i crear un expedient a l'Arxiu del backoffice. Conté una referència a la instància del plugin de l'Arxiu *IArxiuPlugin* que emprarà per fer les diferents crides i moure la informació dels expedients de DISTRIBUCIO cap a un expedient propi. Així mateix, la llibreria *distribucio-*

backoffice-utils.jar utilitza les classes definides en la llibreria client de WS de DISTRIBUCIO *distribucio-ws-client.jar*.

- *BackofficeSistra2Utils*: Aquesta classe d'utilitats té mètodes per tractar els annexos corresponents a documents tècnics de Sistra2 i extreure'n la informació dels formularis o pagaments.

Com que les utilitats de l'Arxiu empen el plugin d'Arxiu la llibreria té dependències del plugin d'Arxiu *plugin-arxiu-api.jar* i de la llibreria de Distribució *distribucio-ws-client.jar*. A continuació es mostra el diagrama de blocs de les llibreries relacionades i el *backoffice* que empra la llibreria d'utilitats.

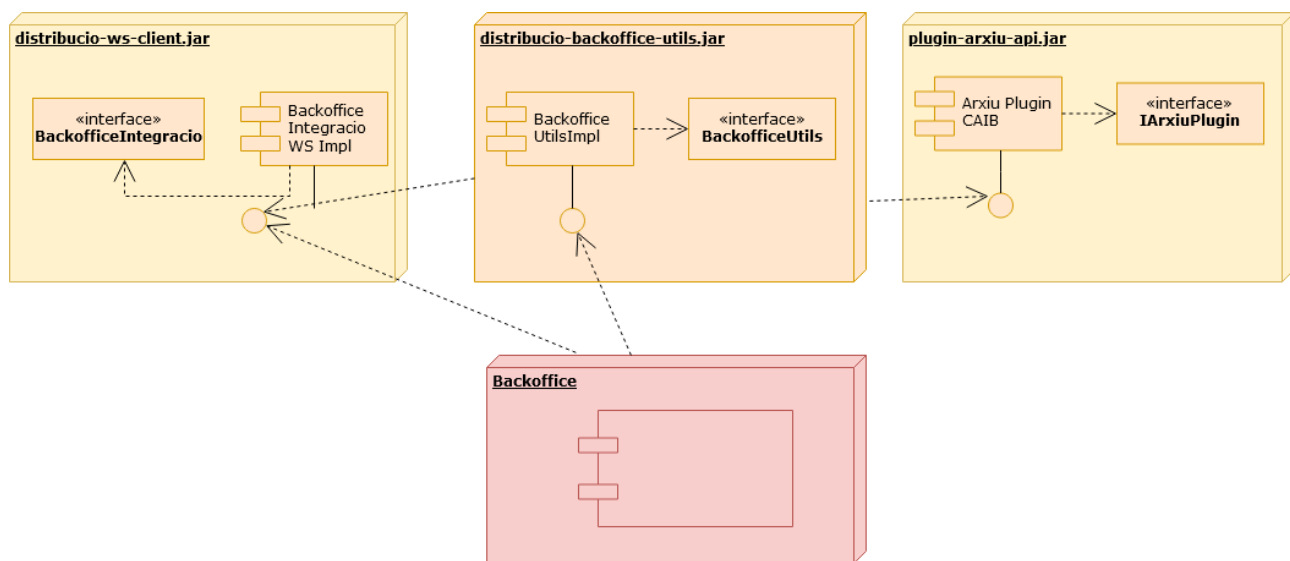


Fig. 1: Diagrama de blocks de la llibreria

4.2. Com incloure la llibreria

La llibreria es pot incloure com una dependència maven dins del codi del backoffice de la mateixa forma que s'inclouen les dependències del client del WS de DISTRIBUCIO o de l' API de l'Arxiu.

S'ha de tenir en compte la versió específica de la llibreria *distribucio-backoffice-utils* ja que aquesta es pot versionar i millorar en el temps. La versió de la llibreria contindrà el número de versió de DISTRIBUCIO i el número de revisió quan aquesta tingui canvis. En el cas de DISTRIBUCIO v. 0.9.26 s'haurà d'incloure la següent dependència *maven*:

```
<dependency>
  <groupId>es.caib.distribucio</groupId>
  <artifactId>distribucio-backoffice-utils</artifactId>
  <version>0.9.28</version>
</dependency>
```


Aquesta llibreria depèn de la llibreria client del WS de DISTRIBUCIO per tractar la informació de DISTRIBUCIO i del plugin d'Arxiu per fer les diferents crides a l'Arxiu.

4.3. Històric

La llibreria està pensada per anar incorporant noves utilitats i en principi se cercarà la compatibilitat cap a versions anteriors. La versió de la llibreria coincideix amb la versió de Distribucio en la que es van introduir els canvis. L'històric de versions actual és el següent:

- 0.9.26 Versió inicial de la llibreria amb la classe *BackofficeUtils* per traspasar arxius.
- 0.9.28 S'afegeix la classe *BackofficeSistra2Utils* per extreure dades de documents tècnics i es reanomena la classe anterior *BackofficeUtils* a *BackofficeArxiuUtils*.

4.4. Utilitats per l'Arxiu

La llibreria d'utilitats inclou la classe *BackofficeArxiuUtils* amb els mètodes per fer les tasques de crear un expedient a l'Arxiu i guardar-hi les dades dels interessats i els arxius de l'anotació de registre. Com que les operacions amb l'Arxiu poden fallar aquesta classe d'Utilitats retorna un objecte amb la informació del resultat de crear l'expedient i de moure els diferents Arxius per a que es puguin fer reintents o es tracti l'error de la manera més escaient. Aquesta classe d'utilitats precisa tenir una referència del plugin d'Arxiu del backoffice per tal de poder operar amb l'Arxiu.

La idea és poder crear i moure la informació de l'expedient a l'Arxiu de DISTRIBUCIO cap a l'expedient a l'Arxiu del *backoffice* a partir de la informació consultada amb el client del WS de DISTRIBUCIO quan el *backoffice* rebí una petició d'anotació pendent. D'aquesta forma la interfície *BackofficeArxiuUtils* defineix el mètode ***crearExpedientAmbAnotacioDeRegistre*** per a fer el seguit de crides necessàries i retornar un resultat que permeti saber com ha anat el resultat.

El mètode *crearExpedientAmbAnotacioRegistre* s'encarregarà, mitjançant la informació proporcionada, de realitzar les següents crides a l'Arxiu:

1. Si no s'informa la propietat o paràmetre *identificador* llavors es crearà el nou expedient a l'arxiu fent una cridada al mètode *expedientCrear* amb la informació de l'expedient passada com a paràmetre. Si es detecta que ja està creat llavors continua.
2. Crear els interessats en l'expedient de l'arxiu fent una cridada al mètode *expedientModificar*. Es comprovarà si l'expedient ja conté els interessats.
3. Crear una nova carpeta a dins l'expedient de destí que contendrà tots els annexos de l'anotació de registre (aquesta passa és opcional i només es fa si hi ha un paràmetre activat). Primer es comprova si la carpeta ja està creada.
4. Moure el document annex des de l'expedient de l'anotació de registre cap al nou expedient creat emprant el mètode *documentMoure*. Com que l'expedient d'origen i el de destí

pertanyen a sèries documentals diferents això provocarà una cridada al servei `dispatchDocument` de l'API REST de l'arxiu. S'ha de comprovar si l'annex s'ha mogut prèviament.

Com que el mètode pot haver produït un error en el transcurs de les crides anteriors internament es faran les comprovacions necessàries en cada pas per rependre l'execució del mètode. Així doncs, el *backoffice* integrador podrà reintentar el processament de l'anotació tornant a invocar el mètode amb els mateixos paràmetres amb l'excepció de la creació de l'expedient, ja que el *backoffice* ha de guardar l'identificador dins l'Arxiu de l'expedient per continuar amb el traspàs de la informació de manera que no es creïn nous expedients en cada intent.

En els següents apartats es descriuen els possibles paràmetres de configuració i es mostra codi d'exemple per al seu ús.

4.4.1. Paràmetres

La classe *BackofficeArxiuUtils* es pot configurar a través dels seus mètodes. Els paràmetres són els següents:

- *iArxiuPlugin*:

És un paràmetre obligatori que s'ha d'especificar a través del constructor *BackofficeArxiuUtilsImpl(IArxiuPlugin)* o a través del mètode *setIArxiuPlugin(IArxiuPlugin)*. Serveix per tenir referència a la instància de l'API de l'Arxiu i poder fer les crides necessàries per a la creació, modificació i consulta de la informació de l'Arxiu.

- *carpeta*:

La definició de la carpeta és opcional. Si es fixa un valor per a la carpeta llavors durant les crides es crearà una carpeta on guardar els annexos de l'anotació de registre. És útil quan l'expedient del *backoffice* pot estar relacionat amb diferents anotacions de registre i es vol separar els annexos de les diferents anotacions. Per fixar el nom de la carpeta es pot fer mitjançant el mètode *setCarpeta(String)*.

- *arxiuPluginListener*:

A fi de poder controlar les diferents crides que es fan a l'Arxiu la interfície *BackofficeArxiuUtils* permet opcionalment fixar una instància que implementi la interfície *ArxiuPluginListener*. Cada cop que es cridi a l'Arxiu s'invocarà al mètode *event* de la instància per a que l'aplicació del *backoffice* pugui monitoritzar la integració amb l'Arxiu amb la informació dels mètodes invocats, els paràmetres més descriptius, si hi ha hagut error i el temps total de processament en ms. El mètode per fixar la instància és *setArxiuPluginListener(ArxiuPluginListener)* i la interfície que ha d'implementar la instància és la següent:

```

package es.caib.distribucio.backoffice.utils;

import java.util.Map;
  
```

```
/** Interfície a implementar si es vol estar informat de les peticions que es fan a
l'Arxiu
*
*/
public interface ArxiuPluginListener {

    /** Mètode per notificar les crides fetes a l'arxiu, els paràmetres, el resultat
i el temps de la petició
    *
    * @param metode           Mètode invocat
    * @param parametres       Paràmetres més representatius
    * @param correcte         Resultat correcte
    * @param error            Descripció de l'error en cas d'error
    * @param e                Excepció en cas de produir-se una excepció
    * @param timeMs           Temps en ms transcorreguts en la crida
    */
    public void event(String metode, Map<String, String> parametres, boolean
correcte, String error, Exception e, long timeMs);
}
```

4.4.2. Ús de *BackofficeArxiuUtils*

Per utilitzar la classe *BackofficeArxiuUtils* amb les utilitats per l'Arxiu s'ha de crear un objecte de la classe *BackofficeArxiuUtilsImpl* amb la referència al plugin d'Arxiu *IPluginArxiu*, consultar el detall de l'anotació de registre mitjançant la llibreria del client de WS de DISTRIBUCIO *distribucio-ws-client* i invocar el mètode ***crearExpedientAmbAnotacioDeRegistre*** amb la informació de l'expedient a crear o a modificar. Si la informació de l'expedient ja conté un identificador d'Arxiu uuid llavors el mètode no en crearà un de nou. Pel contrari, si la informació associada a l'expedient no conté l'identificador dins l'Arxiu llavors el mètode s'encarregarà de crear l'expedient amb les dades passades com a paràmetres. A continuació s'ha de comprovar el resultat de la invocació i en el cas que s'hagi creat l'expedient destí és molt important guardar el seu identificador dins l'Arxiu per poder continuar amb el traspàs d'informació en el cas que no s'hagi pogut traspasar tota la informació dels interessats o dels annexos.

A continuació es mostra el diagrama de classes pel resultat de la crida.

ArxiuResultat	ArxiuResultatAnnex
<ul style="list-style-type: none"> - identificadorExpedient: String - accio: ExpedientAccio - errorCodi: int - errorMessage: String - exception: Exception - resultatAnnexos: List<ArxiuResultatAnnex> 	<ul style="list-style-type: none"> - identificadorAnnex: String - accio: AnnexAccio - errorCodi: int - errorMessage: String - exception: Exception - annex: Annex
<ul style="list-style-type: none"> + getIdentificador() : String + getAccio() + getErrorCodi() : int + getErrorMessage() : String + getException() : Exception + getResultatAnnex(String uuid) : ArxiuResultatAnnex + getResultatAnnexos() : List<ArxiuResultatAnnex> 	<ul style="list-style-type: none"> + getIdentificador() : String + getAccio() + getErrorCodi() : int + getErrorMessage() : String + getException() : Exception + getAnnex() : Annex

Fig. 2: Classes per contenir el resultat de la crida al mètode *crearExpedientAmbAnotacioDeRegistre*.

Si el mètode falla en la creació de l'expedient o el traspàs dels interessats llavors el resultat contindrà l'error i els annexos no s'hauran traspasat.

Quan es traspassin els diferents annexos s'anirà informant el resultat per cada annex. L'usuari de la llibreria podrà consultar el resultat a partir del seu identificador d'Arxiu uuid original a partir del mètode *ArxiuResultat.getResultatAnnex(String uuid)* i obtenir el nou per en l'objecte *ArxiuResultatAnnex* amb el mètode *getIdentificador()*. L'objecte de tipus *ArxiuResultatAnnex* conté l'identificador de l'arxiu si s'ha pogut o existia, l'acció realitzada o la descripció de l'error en el cas que no s'hagi pogut moure.

L'aplicació *backoffice* ha d'implementar la seva lògica pròpia per incorporar la informació dels annexos en el seu propi sistema.

4.4.3. Exemple Java

En el següent exemple de codi es mostra una implemenciació del WS de backoffice de DISTRIBUCIO que rep peticions d'anotacions pendents des de DISTRIBUCIO i les processa amb la llibreria d'utilitats. El codi sencer es pot consultar a l'Error: Reference source not found.

La classe per a l'exemple és la implementació del WS que rep anotacions pendents de registre i té una referència al client del WS d'integració de DISTRIBUCIO per consultar anotacions i una instància del plugin de l'API de l'Arxiu.

taula. Classe de proves per rebre peticions de WS en el mètode *comunicarAnotacionsPendents*.

```
@Component
@WebService(
    name = "Backoffice",
    serviceName = "BackofficeService",
    portName = "BackofficeServicePort",
    endpointInterface = "es.caib.ripea.core.api.service.ws.BackofficeWsServiceBean",
    targetNamespace = "http://www.caib.es/distribucio/ws/backoffice")
public class BackofficeWsServiceImpl implements BackofficeWsService,

    ArxiuPluginListener {

    /** Instància del plugin d'Arxiu */
    private IArxiuPlugin arxiuPlugin = null;

    /** Mètode del WS que rep les comunicacions d'anotacions pendents. */
    @Override
    public void comunicarAnotacionsPendents(List<AnotacioRegistreId> ids) {
        ....
    }
    ...
}
```

Dins del mètode *comunicarAnotacionsPendents* instancia el client del WS d'integració amb el Backoffice i el plugin de l'API de l'Arxiu i consulta l'anotació del registre d'entrada. En la següent porció de codi es pot veure com es crea l'objecte per a la llibreria d'utilitats del backoffice amb la referència al plugin d'Arxiu.

taula. Creació de l'objecte passant l'API d'Arxiu al constructor.

```
// Prepara la cria a la llibreria d'utilitats pel backoffice de DISTRIBUCIO
// Constructor amb la referència al plugin d'Arxiu
BackofficeArxiuUtils backofficeArxiuUtils = new BackofficeArxiuUtilsImpl(getArxiuPlugin());
```

La llibreria permet fixar el nom de la carpeta on es guardaran els annexos i un objecte de tipus *ArxiuPluginListener* si es volen rebre notificacions cada cop que es realitzi una acció d'integració amb l'Arxiu.

taula. Propietat per fixar el nom de la carpeta i l'objecte *ArxiuPluginListener*.

```
// Afegeix la instància de la classe com a escoltador d'events
backofficeArxiuUtils.setArxiuPluginListener(this);
// Estableix la carpeta on guardar els annexos de l'anotació
backofficeArxiuUtils.setCarpeta(anotacio.getIdentificador());
```

En el cas de l'exemple, la classe que implementa el WS també implementa la interfície *ArxiuPluginListener* amb el mètode *event* per obtenir la informació de la crida al plugin de l'API de l'Arxiu.

```
public class BackofficeWsServiceImpl implements BackofficeWsService, ,
    ArxiuPluginListener {
    ...

    /** Mètode que la llibreria del backoffice de distribució crida si es
    fixa una instància que implementa {@link ArxiuPluginListener}.
    *
    */
    @Override
    public void event(
        String metode,
```

```

        Map<String, String> parametres,
        boolean correcte,
        String error,
        Exception e,
        long timeMs) {
    StringBuilder str = new StringBuilder()
        .append("S'ha invocat el mètode \"")
        .append(metode)
        .append("\" amb els paràmetres {");

    int i = 0;
    for (String key : parametres.keySet()) {
        str.append(key).append("=").append(parametres.get(key));
        i++;
        if (i < parametres.size())
            str.append(", ");
    }
    str.append("} amb resultat ").append(correcte ? "OK" : "KO");
    if (error != null)
        str.append(": ").append(error);
    if (e != null)
        str.append(" ").append(e.getClass()).append(" ").append(e.getMessage());
    str.append(" ").append(timeMs).append("ms");

    Logger.debug(str.toString());
}
...
}

```

En el següent exemple de codi es mostra com un cop instanciada la llibreria d'utilitats i consultada l'anotació de registre es crida a la creació de l'expedient amb les dades pròpies de l'expedient. La creació està dins d'un bucle de 10 intents per mostrar com es farien els reintents mentre el codi del resultat sigui diferent de 0.

taula. Crida al mètode *crearExpedientAmbAnotacioRegistre*.

```

// Crida a la creació de l'expedient
String SERIE_DOCUMENTAL = "S0002";
String CLASSIFICACIO = "000000";
ArxiuResultat arxiuResultat;
int intent = 0;
String expedientUuid = null;
do {
    // Crida al mètode de creació de la llibreria
    arxiuResultat = BackofficeArxiuUtils.crearExpedientAmbAnotacioRegistre(
        expedientUuid,
        anotacio.getIdentificador(),
        null,
        Arrays.asList("A04019281"),
        new Date(),
        CLASSIFICACIO,
        ExpedientEstatEnumDto.OBERT,
        SERIE_DOCUMENTAL,
        anotacio);

    expedientUuid = arxiuResultat.getIdentificadorExpedient();
    // Imprimeix el resultat per pantalla
    this.logResultat(
        "Resultat de la crida " + intent++ + " crearExpedientAmbAnotacioRegistre per
l'expedient" + anotacio.getExpedientNumero(),
        arxiuResultat);
} while (arxiuResultat.getErrorCodi() != DistribucioArxiuError.NO_ERROR && intent < 10);

```

A mode informatiu a l'exemple s'imprimeix als logs el contingut de l'objecte *arxiuResultat* amb el mètode *logResultat*.

```
private void logResultat(String descripcio, ArxiuResultat arxiuResultat) {
    Logger.debug(descripcio);
    // Resultat a nivell d'expedient
    Logger.debug("- uuid: " + arxiuResultat.getIdentificadorExpedient());
    Logger.debug("- accio: " + arxiuResultat.getAccio());
    Logger.debug("- errorCodi: " + arxiuResultat.getErrorCodi());
    Logger.debug("- errorMessage: " + arxiuResultat.getErrorMessage());
    Logger.debug("- excepcio: " + (arxiuResultat.getException() != null?
arxiuResultat.getException().getClass() + " " + arxiuResultat.getException().getMessage() : " - "));

    // Resultat pels annexos
    List<ArxiuResultatAnnex> resultatAnnexos = arxiuResultat.getResultatAnnexos();
    Logger.debug(" - Resultat dels " + resultatAnnexos.size() + " annexos:");
    for (ArxiuResultatAnnex resultatAnnex : resultatAnnexos) {
        Logger.debug("\b- uuid: " + resultatAnnex.getIdentificadorAnnex());
        Logger.debug("\b- accio: " + resultatAnnex.getAccio());
        Logger.debug("\b- errorCodi: " + resultatAnnex.getErrorCodi());
        Logger.debug("\b- errorMessage: " + resultatAnnex.getErrorMessage());
        Logger.debug("\b- excepcio: " + (resultatAnnex.getException() != null?
resultatAnnex.getException().getClass() + " " + resultatAnnex.getException().getMessage() : " - "));
    }
}
```

En aquest punt ja s'ha utilitzat la llibreria d'integració per moure continguts i a l'Arxiu s'haurien de poder veure els expedients d'origen i final

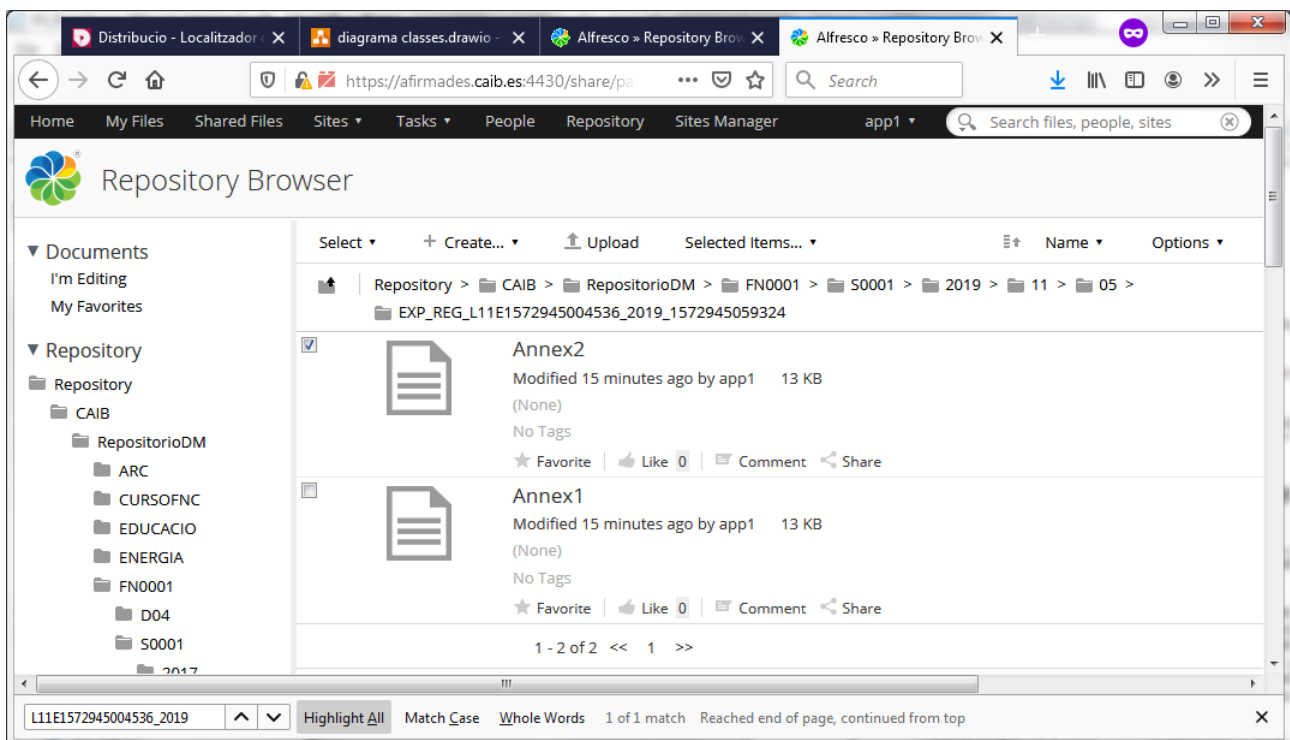


Fig. 3: Expedient origen a l'Arxiu.

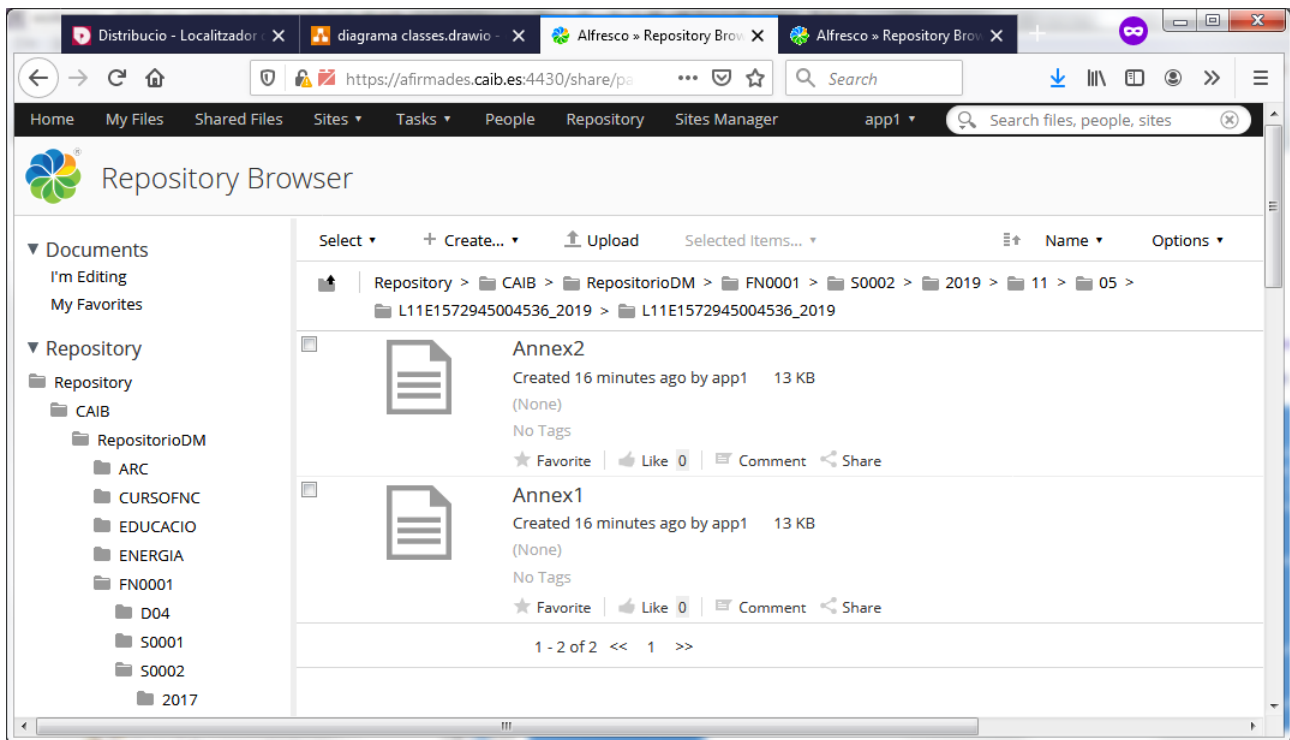


Fig. 4: Expedient després de la crida al mètode de la llibreria d'utilitats.

Per acabar es mostra la porció de codi que consulta el nou expedient i comunica l'estat a l'aplicació de DISTRIBUCIO segons el codi del resultat.

```
// Consultem el nou expedient a l'Arxiu
if (expedientUuid != null) {
    Expedient expedientDetalls =
getArxiuPlugin().expedientDetalls(arxiuResultat.getIdentificadorExpedient(), null);
    logger.debug("S'ha creat l'expedient \" + expedientDetalls.getNom() + \" amb id= \" +
expedientDetalls.getIdentificador() + \" per l'anotació \" + anotacio.getIdentificador() + \" amb els següents
continguts:");
    for (ContingutArxiu contingut : expedientDetalls.getContinguts()) {
        logger.debug("- \" + contingut.getIdentificador() + \" \" + contingut.getNom() + \" amb \" +
(contingut.getFirmes() != null? contingut.getFirmes().size() : 0) + \" firmes.");
    }
} else {
    logger.warn("L'expedient no s'ha pogut crear per l'anotació \" + anotacio.getIdentificador());
}
// Es comunica el resultat a DISTRIBUCIO
switch(arxiuResultat.getErrorCodi()) {
    case 0:
        backofficeClient.canviEstat(idWs,
es.caib.distribucio.ws.backofficeintegracio.Estat.PROCESSADA, "Processada");
        break;
    default:
        backofficeClient.canviEstat(idWs, es.caib.distribucio.ws.backofficeintegracio.Estat.ERROR,
arxiuResultat.getErrorCodi() + \" \" + arxiuResultat.getErrorMessage());
        break;
}
```

4.5. Utilitats per Sistra2

Serà un cas comú que els tràmits telemàtics de Sistra2 arribin als backoffices de Distribucio amb documents tècnics de pagaments o formularis com annexos. Aquests documents contenen un XML

definit pels esquemes XSD corresponents de Sistra2 i contenen les dades d'un formulari o d'un pagament. A fi de facilitar el la interpretació del contingut per part dels backoffices de Distribució s'ha creat la classe d'utilitats *BackofficeSistra2Utils* amb els mètodes per extreure la informació dels documents tècnics XML. En aquest apartat es descriu l'ús de la llibreria per aquest cas i un exemple Java.

4.5.1. Ús de *BackofficeSistra2Utils*

Per utilitzar la classe *BackofficeSistra2Utils* amb les utilitats per Sistra2 s'ha de crear un objecte de la classe *BackofficeSistra2UtilsImpl*. Aquesta classe no precisa de configurar i conté 2 mètodes per llegir el contingut d'un document annex i retornar la informació d'un formulari o d'un pagament. Els mètodes que conté són:

- *Formulario* *parseXmlFormulario(byte[])* : Processa el contingut en bytes del document annex del formulari i retorna les dades del pagament en forma d'objecte *Formulario*.
- *Pago* *parseXmlPago(byte[])* : Processa el contingut en bytes del document annex del pagament i retorna les dades del pagament en forma d'objecte *Pago*.

Tant la classe *Formulario* com la classe *Pago* estan generades a partir dels esquemes XSD de les dades XML provinents de Sistra i els mètodes serveixen per transformar les dades XML a objectes que puguin ser tractats pel backoffice.

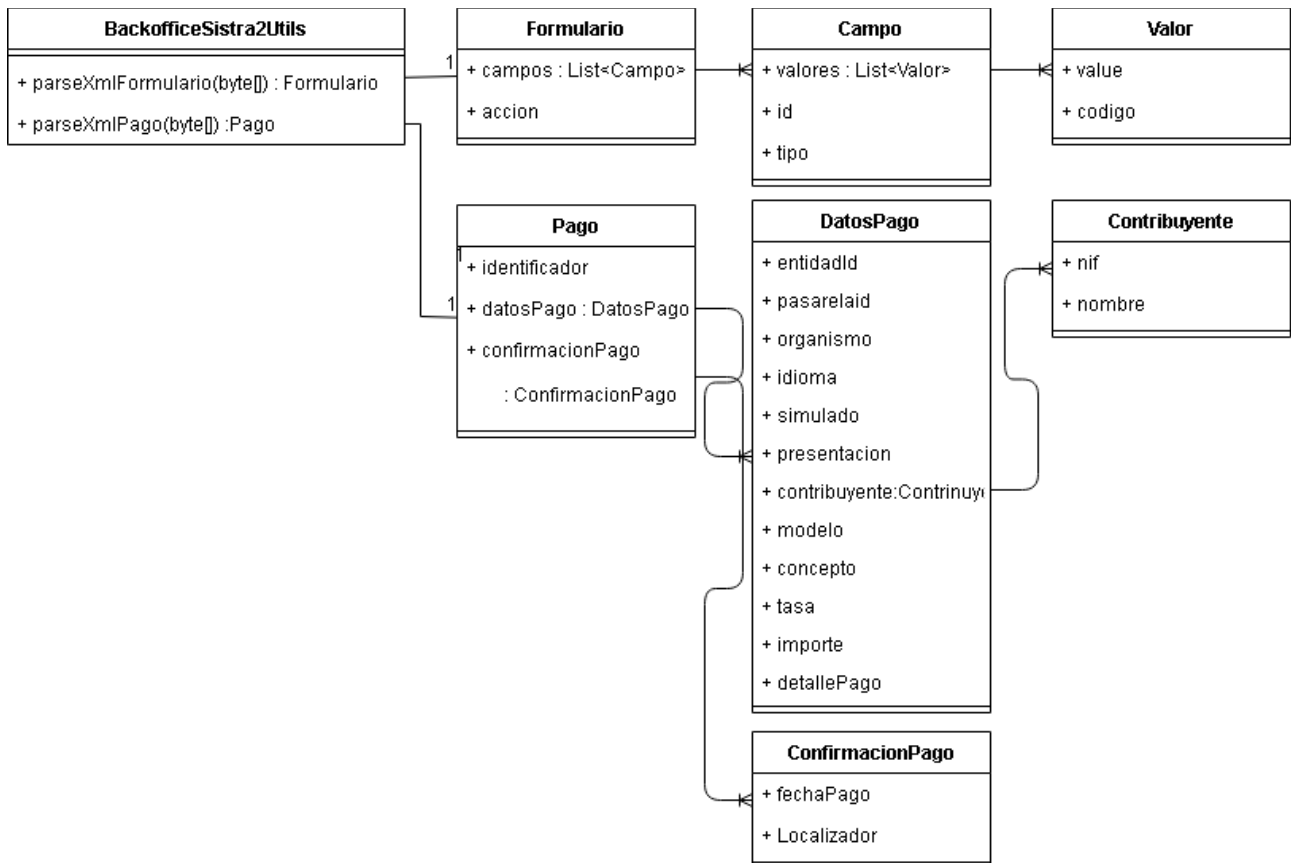


Fig. 5: Diagrama de classes de *BackofficeSistra2Utils* i les classes de *Formulario* i *Pago* que retorna.

Els mètodes reben el contingut en *byte[]* com a paràmetre d'entrada i retornen un objecte *Formulario* o *Pago* segons el mètode invocat. Cal controlar qualsevol excepció que es pugui produir quan el contingut de l'annex no és vàlid i el parseig de l'XML falla.

4.5.2. Exemple Java

En el següent exemple de codi es mostra un exemple de crida al mètode de conversió de l'annex a dades de formulari.

taula. Conversió d'un annex corresponent a un document tècnic d'un formulari XML.

```
// Inicialitza la classe d'utilitats de Sistra2
BackofficeSistra2Utils sistraUtils = new BackofficeSistra2UtilsImpl();

// Simula l'annex
byte[] contingut =
Files.readAllBytes(Paths.get(this.getClass().getClassLoader().getResource("formulario.xml").toURI()));
// Llegeix el formulari
Formulario formulario = null;
try {
    formulario = sistraUtils.parseXmlFormulario(contingut);
} catch (Exception e) {
    throw new Exception("Error llegint l'annex : " + e.getMessage());
}
System.out.println("Llegit el formmlari, formulario.accion=" + formulario.getAccion());
```

Per a l'exemple s'ha creat l'arxiu *formulario.xml* amb l' XML d'exemple per a la lectura. A continuació es mostra aquest XML d'exemple.

taula. Exemple d' XML pel formulari.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:FORMULARIO accion="accion" xmlns:ns2="urn:es:caib:sistra2:xml:formulario:v1:model">
  <CAMPO id="campo_0">
    <VALOR codigo="cod_0">valor_i</VALOR>
    <VALOR codigo="cod_0">valor_i</VALOR>
  </CAMPO>
  <CAMPO id="campo_1">
    <VALOR codigo="cod_1">valor_i</VALOR>
    <VALOR codigo="cod_1">valor_i</VALOR>
  </CAMPO>
</ns2:FORMULARIO>
```

Annex I: WSDL del servei d'enviament a bústies

```
< xs:element name = "enviarAnotacioRegistreEntradaResponse"
    type = "tns:enviarAnotacioRegistreEntradaResponse" />
< xs:element name = "enviarDocument"
    type = "tns:enviarDocument" />
< xs:element name = "enviarDocumentResponse"
    type = "tns:enviarDocumentResponse" />
< xs:element name = "enviarExpedient"
    type = "tns:enviarExpedient" />
< xs:element name = "enviarExpedientResponse"
    type = "tns:enviarExpedientResponse" />
< xs:complexType name = "enviarDocument" >
    < xs:sequence >
        < xs:element name = "entitat" type = "xs:string" />
        < xs:element name = "unitatAdministrativa" type =
"xs:string" />
        < xs:element name = "referenciaDocument" type =
"xs:string" />
    </ xs:sequence >
</ xs:complexType >
< xs:complexType name = "enviarDocumentResponse" >
    < xs:sequence />
</ xs:complexType >
< xs:complexType name = "enviarExpedient" >
    < xs:sequence >
        < xs:element name = "entitat" type = "xs:string" />
        < xs:element name = "unitatAdministrativa" type =
"xs:string" />
        < xs:element name = "referenciaExpedient" type =
"xs:string" />
    </ xs:sequence >
</ xs:complexType >
< xs:complexType name = "enviarExpedientResponse" >
    < xs:sequence />
</ xs:complexType >
< xs:complexType name = "enviarAnotacioRegistreEntrada" >
    < xs:sequence >
        < xs:element name = "entitat" type = "xs:string" />
        < xs:element name = "unitatAdministrativa" type =
"xs:string" />
        < xs:element name = "registreEntrada"
            type = "tns:registreAnotacio" />
    </ xs:sequence >
</ xs:complexType >
```

```

        </ xs:sequence >
    </ xs:complexType >
    < xs:complexType name = "registreAnotacio" >
        < xs:sequence >
            < xs:element maxOccurs = "unbounded" minOccurs = "0"
                name = "annexos" nillable = "true" type =
"tns:registreAnnex" />

            < xs:element minOccurs = "0" name = "aplicacioCodi"
                type = "xs:string" />
            < xs:element minOccurs = "0" name =
"aplicacioVersio"

                type = "xs:string" />
            < xs:element minOccurs="0" name="procedimentCodi"
                type="xs:string" />
            < xs:element name="presencial"
                type="xs:boolean" />
            < xs:element minOccurs = "0" name = "assumptecodi"
                type = "xs:string" />
            < xs:element minOccurs = "0" name =
"assumptedescripcio"

                type = "xs:string" />
            < xs:element minOccurs = "0" name =
"assumptetipuscodi"

                type = "xs:string" />
            < xs:element minOccurs = "0" name =
"assumptetipusdescripcio"

                type = "xs:string" />
            < xs:element minOccurs = "0" name = "data" type =
"xs:dateTime" />

            < xs:element minOccurs = "0" name = "dataorigen"
                type = "xs:dateTime" />
            < xs:element minOccurs = "0" name =
"documentaciofisicacodi"

                type = "xs:string" />
            < xs:element minOccurs = "0"
                name = "documentaciofisicaDescripcio" type =
"xs:string" />

            < xs:element minOccurs = "0" name = "entitatCodi"
                type = "xs:string" />
            < xs:element minOccurs = "0" name =
"entitatdescripcio"

                type = "xs:string" />
            < xs:element minOccurs = "0" name =
"expedientNumero"

                type = "xs:string" />
            < xs:element minOccurs = "0" name = "exposa" type =
"xs:string" />

```

```

< xs:element minOccurs = "0" name = "extracte"
    type = "xs:string" />
< xs:element minOccurs = "0" name = "identificador"
    type = "xs:string" />
< xs:element minOccurs = "0" name = "idiomaCodi"
    type = "xs:string" />
< xs:element minOccurs = "0" name =
"idiomaDescripcio"
    type = "xs:string" />
< xs:element maxOccurs = "unbounded" minOccurs = "0"
    name = "interessats" nillable = "true" type =
"tns:registreInteressat" />
< xs:element minOccurs = "0" name = "justificant"
    type = "tns:registreAnnex" />
< xs:element minOccurs = "0" name = "llibreCodi"
    type = "xs:string" />
< xs:element minOccurs = "0" name =
"llibreDescripcio"
    type = "xs:string" />
< xs:element minOccurs = "0" name = "numero" type =
"xs:string" />
< xs:element minOccurs = "0" name = "numeroOrigen"
    type = "xs:string" />
< xs:element minOccurs = "0" name = "observacions"
    type = "xs:string" />
< xs:element minOccurs = "0" name = "oficinaCodi"
    type = "xs:string" />
< xs:element minOccurs = "0" name =
"oficinaDescripcio"
    type = "xs:string" />
< xs:element minOccurs = "0" name =
"oficinaOrigenCodi"
    type = "xs:string" />
< xs:element minOccurs = "0" name =
"oficinaOrigenDescripcio"
    type = "xs:string" />
< xs:element minOccurs = "0" name = "referencia"
    type = "xs:string" />
< xs:element minOccurs = "0" name = "solicita"
    type = "xs:string" />
<xs:element minOccurs="0" name="tipusES"
type="xs:string" />
< xs:element minOccurs = "0" name =
"transportNumero"
    type = "xs:string" />
< xs:element minOccurs = "0" name =
"transportTipusCodi"

```

```

        type = "xs:string" />
      < xs:element minOccurs = "0"
        name = "transportTipusDescripcio" type =
"xs:string" />

      < xs:element minOccurs = "0" name = "usuariCodi"
        type = "xs:string" />
      < xs:element minOccurs = "0" name = "usuariContacte"
        type = "xs:string" />
      < xs:element minOccurs = "0" name = "usuariNom"
        type = "xs:string" />
    </ xs:sequence >
  </ xs:complexType >
  < xs:complexType name = "registreAnnex" >
    < xs:sequence >
      < xs:element minOccurs = "0" name = "eniDataCaptura"
        type = "xs:dateTime" />
      < xs:element minOccurs = "0" name =
"eniEstatElaboracio"
        type = "xs:string" />
      < xs:element minOccurs = "0" name = "eniOrigen"
        type = "xs:string" />
      < xs:element minOccurs = "0" name =
"eniTipusDocumental"
        type = "xs:string" />
      < xs:element maxOccurs = "unbounded" minOccurs = "0"
        name = "firmes" nillable = "true" type =
"tns:firma" />
      < xs:element minOccurs = "0" name =
"fitxerArxiuUuid"
        type = "xs:string" />
      < xs:element minOccurs = "0" name =
"fitxerContingut"
        type = "xs:base64Binary" />
      < xs:element minOccurs = "0" name = "fitxerNom"
        type = "xs:string" />
      < xs:element name = "fitxerTamany" type = "xs:int"
    />
      < xs:element minOccurs = "0" name =
"fitxerTipusMime"
        type = "xs:string" />
      < xs:element minOccurs = "0" name = "id" type =
"xs:long" />
      < xs:element minOccurs = "0" name = "localitzacio"
        type = "xs:string" />
      < xs:element name = "metaDades"
    < xs:complexType>
    < xs:sequence>

```

```

<xs:element maxOccurs="unbounded" minOccurs="0"
  name="entry">
  <xs:complexType>
  <xs:sequence>
  <xs:element minOccurs="0" name="key"
type="xs:string" />
    <xs:element minOccurs="0" name="value"
type="xs:string" />
  </xs:sequence>
  </xs:complexType>
  </xs:element>
  </xs:sequence>
  </xs:complexType>
  </xs:element>
  <xs:element minOccurs="0" name="observacions"
    type="xs:string" />
  <xs:element minOccurs="0" name="
"sicresTipusDocument"
    type="xs:string" />
  <xs:element minOccurs="0" name="timestamp"
    type="xs:string" />
  <xs:element minOccurs="0" name="titol" type="
"xs:string" />
    <xs:element minOccurs="0" name="validacioOCSP"
    type="xs:string" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="firma">
  <xs:sequence>
    <xs:element minOccurs="0" name="contingut"
      type="xs:base64Binary" />
    <xs:element minOccurs="0" name="csv" type="
"xs:string" />
    <xs:element minOccurs="0" name="csvRegulacio"
      type="xs:string" />
    <xs:element minOccurs="0" name="fitxerNom"
      type="xs:string" />
    <xs:element minOccurs="0" name="perfil" type="
"xs:string" />
    <xs:element minOccurs="0" name="tipus" type="
"xs:string" />
    <xs:element minOccurs="0" name="tipusMime"
      type="xs:string" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="registreInteressat">
  <xs:sequence>
    <xs:element minOccurs="0" name="adresa" type="
"xs:string" />

```

```

< xs:element minOccurs = "0" name = "canalPreferent"
    type = "xs:string" />
< xs:element minOccurs = "0" name = "codiDire"
    type = "xs:string" />
< xs:element minOccurs = "0" name = "codiPostal"
    type = "xs:string" />
< xs:element minOccurs = "0" name = "documentNum"
    type = "xs:string" />
< xs:element minOccurs = "0" name = "documentTipus"
    type = "xs:string" />
< xs:element minOccurs = "0" name = "email" type =
"xs:string" />

< xs:element minOccurs = "0" name = "emailHabilitat"
    type = "xs:string" />
< xs:element minOccurs = "0" name = "l·linatgel"
    type = "xs:string" />
< xs:element minOccurs = "0" name = "l·linatge2"
    type = "xs:string" />
< xs:element minOccurs = "0" name = "municipi"
    type = "xs:string" />
< xs:element minOccurs = "0" name = "municipiCodi"
    type = "xs:string" />
< xs:element minOccurs = "0" name = "nom" type =
"xs:string" />

< xs:element minOccurs = "0" name = "observacions"
    type = "xs:string" />
< xs:element minOccurs = "0" name = "pais" type =
"xs:string" />

< xs:element minOccurs = "0" name = "paisCodi" type =
"xs:string" />

< xs:element minOccurs = "0" name = "provincia"
    type = "xs:string" />
< xs:element minOccurs = "0" name = "provinciaCodi"
    type = "xs:string" />
< xs:element minOccurs = "0" name = "raoSocial"
    type = "xs:string" />
< xs:element minOccurs = "0" name = "representant"
    type = "tns:registreInteressat" />
< xs:element minOccurs = "0" name = "telefon"
    type = "xs:string" />
< xs:element minOccurs = "0" name = "tipus" type =
"xs:string" />

    </ xs:sequence >
  </ xs:complexType >
  < xs:complexType
    name = "enviarAnotacioRegistreEntradaResponse" >

```

```

        < xs:sequence />
      </ xs:complexType >
    </ xs:schema >
  </ wsdl:types >
  < wsdl:message name = "enviarDocument" >
    < wsdl:part element = "tns:enviarDocument" name = "parameters" >
    </ wsdl:part >
  </ wsdl:message >
  < wsdl:message name = "enviarAnotacioRegistreEntrada" >
    < wsdl:part element = "tns:enviarAnotacioRegistreEntrada"
      name = "parameters" >
    </ wsdl:part >
  </ wsdl:message >
  < wsdl:message name = "enviarExpedientResponse" >
    < wsdl:part element = "tns:enviarExpedientResponse"
      name = "parameters" >
    </ wsdl:part >
  </ wsdl:message >
  < wsdl:message name = "enviarExpedient" >
    < wsdl:part element = "tns:enviarExpedient" name = "parameters" >
    </ wsdl:part >
  </ wsdl:message >
  < wsdl:message name = "enviarAnotacioRegistreEntradaResponse" >
    < wsdl:part
      element = "tns:enviarAnotacioRegistreEntradaResponse" name =
"parameters" >
    </ wsdl:part >
  </ wsdl:message >
  < wsdl:message name = "enviarDocumentResponse" >
    < wsdl:part element = "tns:enviarDocumentResponse"
      name = "parameters" >
    </ wsdl:part >
  </ wsdl:message >
  < wsdl:portType name = "BustiaV1" >
    < wsdl:operation name = "enviarDocument" >
      < wsdl:input message = "tns:enviarDocument"
        name = "enviarDocument" >
      </ wsdl:input >
      < wsdl:output message = "tns:enviarDocumentResponse"
        name = "enviarDocumentResponse" >
      </ wsdl:output >
    </ wsdl:operation >
    < wsdl:operation name = "enviarExpedient" >
      < wsdl:input message = "tns:enviarExpedient"
        name = "enviarExpedient" >
      </ wsdl:input >

```



```

    < wsdl:output message = "tns:enviarExpedientResponse"
      name = "enviarExpedientResponse" >
    </ wsdl:output >
  </ wsdl:operation >
  < wsdl:operation name = "enviarAnotacioRegistreEntrada" >
    < wsdl:input message = "tns:enviarAnotacioRegistreEntrada"
      name = "enviarAnotacioRegistreEntrada" >
    </ wsdl:input >
    < wsdl:output
      message = "tns:enviarAnotacioRegistreEntradaResponse"
      name = "enviarAnotacioRegistreEntradaResponse" >
    </ wsdl:output >
  </ wsdl:operation >
</ wsdl:portType >
< wsdl:binding name = "BustiaV1ServiceSoapBinding"
  type = "tns:BustiaV1" >
  < soap:binding style = "document"
    transport = "http://schemas.xmlsoap.org/soap/http" />
  < wsdl:operation name = "enviarDocument" >
    < soap:operation soapAction = "" style = "document" />
    < wsdl:input name = "enviarDocument" >
      < soap:body use = "literal" />
    </ wsdl:input >
    < wsdl:output name = "enviarDocumentResponse" >
      < soap:body use = "literal" />
    </ wsdl:output >
  </ wsdl:operation >
  < wsdl:operation name = "enviarExpedient" >
    < soap:operation soapAction = "" style = "document" />
    < wsdl:input name = "enviarExpedient" >
      < soap:body use = "literal" />
    </ wsdl:input >
    < wsdl:output name = "enviarExpedientResponse" >
      < soap:body use = "literal" />
    </ wsdl:output >
  </ wsdl:operation >
  < wsdl:operation name = "enviarAnotacioRegistreEntrada" >
    < soap:operation soapAction = "" style = "document" />
    < wsdl:input name = "enviarAnotacioRegistreEntrada" >
      < soap:body use = "literal" />
    </ wsdl:input >
    < wsdl:output name = "enviarAnotacioRegistreEntradaResponse" >
      < soap:body use = "literal" />
    </ wsdl:output >
  </ wsdl:operation >
</ wsdl:binding >

```

```
< wsdl:service name = "BustiaV1Service" >
  < wsdl:port binding = "tns:BustiaV1ServiceSoapBinding"
    name = "BustiaV1ServicePort" >
    < soap:address
      location =
"http://localhost:8080/distribucio/ws/v1/bustia" />
    </ wsdl:port >
  </ wsdl:service >
</ wsdl:definitions >
```

Annex II: WSDL del backoffice

```
<? xml version = '1.0' encoding = 'UTF-8' ?>< wsdl:definitions name = "BackofficeService"
targetNamespace = "http://www.caib.es/distribucio/ws/backoffice" xmlns:ns1 =
"http://schemas.xmlsoap.org/wsdl/soap/http" xmlns:soap =
"http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns =
"http://www.caib.es/distribucio/ws/backoffice" xmlns:wsdl =
"http://schemas.xmlsoap.org/wsdl/" xmlns:xsd = "http://www.w3.org/2001/XMLSchema" >
  < wsdl:types >
    < xs:schema elementFormDefault = "unqualified" targetNamespace =
"http://www.caib.es/distribucio/ws/backoffice" version = "1.0" xmlns:tns =
"http://www.caib.es/distribucio/ws/backoffice" xmlns:xs =
"http://www.w3.org/2001/XMLSchema" >
      < xs:element name = "comunicarAnotacionsPendents" type =
"tns:comunicarAnotacionsPendents" />
      < xs:element name = "comunicarAnotacionsPendentsResponse" type =
"tns:comunicarAnotacionsPendentsResponse" />
      < xs:complexType name = "comunicarAnotacionsPendents" >
        < xs:sequence >
          < xs:element maxOccurs = "unbounded" name = "ids" type =
"tns:anotacioRegistreId" />
        </ xs:sequence >
      </ xs:complexType >
      < xs:complexType name = "anotacioRegistreId" >
        < xs:sequence >
          < xs:element minOccurs = "0" name = "clauAcces" type = "xs:string" />
          < xs:element minOccurs = "0" name = "indetificador" type = "xs:string" />
        </ xs:sequence >
      </ xs:complexType >
      < xs:complexType name = "comunicarAnotacionsPendentsResponse" >
        < xs:sequence />
      </ xs:complexType >
    </ xs:schema >
  </ wsdl:types >
  < wsdl:message name = "comunicarAnotacionsPendents" >
    < wsdl:part element = "tns:comunicarAnotacionsPendents" name =
"parameters" >
```

```

    </ wsdl:part >
  </ wsdl:message >
  < wsdl:message name = "comunicarAnotacionsPendentsResponse" >
    < wsdl:part element = "tns:comunicarAnotacionsPendentsResponse" name =
"parameters" >
      </ wsdl:part >
    </ wsdl:message >
    < wsdl:portType name = "Backoffice" >
      < wsdl:operation name = "comunicarAnotacionsPendents" >
        < wsdl:input message = "tns:comunicarAnotacionsPendents" name =
"comunicarAnotacionsPendents" >
          </ wsdl:input >
          < wsdl:output message = "tns:comunicarAnotacionsPendentsResponse"
name = "comunicarAnotacionsPendentsResponse" >
            </ wsdl:output >
          </ wsdl:operation >
        </ wsdl:portType >
        < wsdl:binding name = "BackofficeServiceSoapBinding" type =
"tns:Backoffice" >
          < soap:binding style = "document" transport =
"http://schemas.xmlsoap.org/soap/http" />
          < wsdl:operation name = "comunicarAnotacionsPendents" >
            < soap:operation soapAction = "" style = "document" />
            < wsdl:input name = "comunicarAnotacionsPendents" >
              < soap:body use = "literal" />
            </ wsdl:input >
            < wsdl:output name = "comunicarAnotacionsPendentsResponse" >
              < soap:body use = "literal" />
            </ wsdl:output >
          </ wsdl:operation >
        </ wsdl:binding >
        < wsdl:service name = "BackofficeService" >
          < wsdl:port binding = "tns:BackofficeServiceSoapBinding" name =
"BackofficeServicePort" >
            < soap:address location =
"http://localhost:8080/distribucio/ws/backoffice" />
          </ wsdl:port >
        </ wsdl:service >
      </ wsdl:definitions >

```

Annex III: WSDL del servei d'integració

```
<?xml version='1.0' encoding='UTF-8'?>
<wsdl:definitions name="BackofficeIntegracioService"
  targetNamespace="http://www.caib.es/distribucio/ws/backofficeIntegracio"
  xmlns:ns1="http://schemas.xmlsoap.org/wsdl/soap/http"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tns="http://www.caib.es/distribucio/ws/backofficeIntegracio"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <wsdl:types>
    <xs:schema elementFormDefault="unqualified"

      targetNamespace="http://www.caib.es/distribucio/ws/backofficeIntegracio"
      version="1.0"
      xmlns:tns="http://www.caib.es/distribucio/ws/backofficeIntegracio"
      xmlns:xs="http://www.w3.org/2001/XMLSchema">
        <xs:element name="canviEstat" type="tns:canviEstat" />
        <xs:element name="canviEstatResponse"
type="tns:canviEstatResponse" />
        <xs:element name="consulta" type="tns:consulta" />
        <xs:element name="consultaResponse" type="tns:consultaResponse" />
        <xs:complexType name="consulta">
          <xs:sequence>
            <xs:element name="id" type="tns:anotacioRegistreId"
/>

          </xs:sequence>
        </xs:complexType>
        <xs:complexType name="anotacioRegistreId">
          <xs:sequence>
            <xs:element minOccurs="0" name="clauAcces"
type="xs:string" />
            <xs:element minOccurs="0" name="indetificador"
type="xs:string" />
          </xs:sequence>
        </xs:complexType>
        <xs:complexType name="consultaResponse">
          <xs:sequence>
            <xs:element minOccurs="0" name="return"
type="tns:anotacioRegistreEntrada" />
          </xs:sequence>
        </xs:complexType>
        <xs:complexType name="anotacioRegistreEntrada">
          <xs:complexContent>
            <xs:extension base="tns:anotacioRegistreBase">
              <xs:sequence>
                <xs:element minOccurs="0"
name="destiCodi" type="xs:string" />
                <xs:element minOccurs="0"
name="destiDescripcio" type="xs:string" />
              </xs:sequence>
            </xs:extension>
          </xs:complexContent>
        </xs:complexType>
      </xs:schema>
    </wsdl:types>
</wsdl:definitions>
```

```

</xs:complexType>
<xs:complexType name="anotacioRegistreBase">
  <xs:sequence>
    <xs:element maxOccurs="unbounded" minOccurs="0"
name="annexos"
nillable="true" type="tns:annex" />
    <xs:element minOccurs="0" name="aplicacioCodi"
type="xs:string" />
    <xs:element minOccurs="0" name="aplicacioVersio"
type="xs:string" />
    <xs:element minOccurs="0" name="assumppteCodiCodi"
type="xs:string" />
    <xs:element minOccurs="0"
name="assumppteCodiDescripcio"
type="xs:string" />
    <xs:element minOccurs="0" name="assumppteTipusCodi"
type="xs:string" />
    <xs:element minOccurs="0"
name="assumppteTipusDescripcio"
type="xs:dateTime" />
    <xs:element minOccurs="0" name="data"
type="xs:string" />
    <xs:element minOccurs="0" name="docFisicaCodi"
type="xs:string" />
    <xs:element minOccurs="0" name="docFisicaDescripcio"
type="xs:string" />
    <xs:element minOccurs="0" name="entitatCodi"
type="xs:string" />
    <xs:element minOccurs="0" name="entitatDescripcio"
type="xs:string" />
    <xs:element minOccurs="0" name="expedientNumero"
type="xs:string" />
    <xs:element minOccurs="0" name="exposa"
type="xs:string" />
    <xs:element minOccurs="0" name="extracte"
type="xs:string" />
    <xs:element minOccurs="0" name="identificador"
type="xs:string" />
    <xs:element minOccurs="0" name="idiomaCodi"
type="xs:string" />
    <xs:element minOccurs="0" name="idomaDescripcio"
type="xs:string" />
    <xs:element maxOccurs="unbounded" minOccurs="0"
name="interessats" nillable="true"
type="tns:interessat" />
    <xs:element minOccurs="0" name="llibreCodi"
type="xs:string" />
    <xs:element minOccurs="0" name="llibreDescripcio"
type="xs:string" />
    <xs:element minOccurs="0" name="observacions"
type="xs:string" />
    <xs:element minOccurs="0" name="oficinaCodi"
type="xs:string" />
    <xs:element minOccurs="0" name="oficinaDescripcio"
type="xs:string" />
    <xs:element minOccurs="0" name="origenData"
type="xs:dateTime" />
    <xs:element minOccurs="0" name="origenRegistreNumero"

```

```

type="xs:string" />
type="xs:string" />
type="xs:string" />
type="xs:string" />
name="transportTipusDescripcio"
type="xs:string" />
type="xs:string" />
</xs:sequence>
</xs:complexType>
<xs:complexType name="annex">
  <xs:sequence>
    <xs:element minOccurs="0" name="contingut"
type="xs:base64Binary" />
    <xs:element minOccurs="0" name="firmaContingut"
type="xs:base64Binary" />
    <xs:element minOccurs="0" name="firmaNom"
type="xs:string" />
    <xs:element minOccurs="0" name="firmaPerfil"
type="tns:firmaPerfil" />
    <xs:element name="firmaTamany" type="xs:long" />
    <xs:element minOccurs="0" name="firmaTipus"
type="tns:firmaTipus" />
    <xs:element minOccurs="0" name="firmaTipusMime"
type="xs:string" />
    <xs:element minOccurs="0" name="nom" type="xs:string"
/>
    <xs:element minOccurs="0" name="ntiEstadoElaboracion"
type="tns:ntiEstadoElaboracion" />
    <xs:element minOccurs="0" name="ntiFechaCaptura"
type="xs:dateTime" />
    <xs:element minOccurs="0" name="ntiOrigen"
type="tns:ntiOrigen" />
    <xs:element minOccurs="0" name="ntiTipoDocumental"
type="tns:ntiTipoDocumento" />
    <xs:element minOccurs="0" name="observacions"
type="xs:string" />
    <xs:element minOccurs="0" name="sicresTipoDocumento"
type="tns:sicresTipoDocumento" />
    <xs:element minOccurs="0"
name="sicresValidezDocumento"
type="tns:sicresValidezDocumento" />
    <xs:element name="tamany" type="xs:long" />
    <xs:element minOccurs="0" name="tipusMime"
type="xs:string" />
    <xs:element minOccurs="0" name="titol"
type="xs:string" />
    <xs:element minOccurs="0" name="uuid"

```

```

type="xs:string" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="interessat">
    <xs:complexContent>
      <xs:extension base="tns:interessatBase">
        <xs:sequence>
          <xs:element minOccurs="0"
name="representant" type="tns:representant" />
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="interessatBase">
    <xs:sequence>
      <xs:element minOccurs="0" name="adresa"
type="xs:string" />
      <xs:element minOccurs="0" name="adresaElectronica"
type="xs:string" />
      <xs:element minOccurs="0" name="canal"
type="xs:string" />
      <xs:element minOccurs="0" name="cp" type="xs:string" />
      <xs:element minOccurs="0" name="documentNumero"
type="xs:string" />
      <xs:element minOccurs="0" name="documentTipus"
type="tns:documentTipus" />
      <xs:element minOccurs="0" name="email"
type="xs:string" />
      <xs:element minOccurs="0" name="l·linatge1"
type="xs:string" />
      <xs:element minOccurs="0" name="l·linatge2"
type="xs:string" />
      <xs:element minOccurs="0" name="municipi"
type="xs:string" />
      <xs:element minOccurs="0" name="municipiCodi"
type="xs:string" />
      <xs:element minOccurs="0" name="nom" type="xs:string" />
      <xs:element minOccurs="0" name="observacions"
type="xs:string" />
      <xs:element minOccurs="0" name="organCodi"
type="xs:string" />
      <xs:element minOccurs="0" name="pais"
type="xs:string" />
      <xs:element minOccurs="0" name="paisCodi"
type="xs:string" />
      <xs:element minOccurs="0" name="provincia"
type="xs:string" />
      <xs:element minOccurs="0" name="provinciaCodi"
type="xs:string" />
      <xs:element minOccurs="0" name="raoSocial"
type="xs:string" />
      <xs:element minOccurs="0" name="telefon"
type="xs:string" />
      <xs:element minOccurs="0" name="tipus"
type="tns:interessatTipus" />
    </xs:sequence>
  
```



```

</xs:complexType>
<xs:complexType name="representant">
  <xs:complexContent>
    <xs:extension base="tns:interessatBase">
      <xs:sequence />
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="canviEstat">
  <xs:sequence>
    <xs:element name="id" type="tns:anotacioRegistreId"
/>
    <xs:element name="estat" type="tns:estat" />
    <xs:element minOccurs="0" name="observacions"
type="xs:string" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="canviEstatResponse">
  <xs:sequence />
</xs:complexType>
<xs:simpleType name="firmaPerfil">
  <xs:restriction base="xs:string">
    <xs:enumeration value="BES" />
    <xs:enumeration value="EPES" />
    <xs:enumeration value="LTV" />
    <xs:enumeration value="T" />
    <xs:enumeration value="C" />
    <xs:enumeration value="X" />
    <xs:enumeration value="XL" />
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="firmaTipus">
  <xs:restriction base="xs:string">
    <xs:enumeration value="CSV" />
    <xs:enumeration value="XADES_DET" />
    <xs:enumeration value="XADES_ENV" />
    <xs:enumeration value="CADES_DET" />
    <xs:enumeration value="CADES_ATT" />
    <xs:enumeration value="PADES" />
    <xs:enumeration value="SMIME" />
    <xs:enumeration value="ODT" />
    <xs:enumeration value="OOXML" />
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="ntiEstadoElaboracion">
  <xs:restriction base="xs:string">
    <xs:enumeration value="ORIGINAL" />
    <xs:enumeration
value="COPIA_ELECT_AUTENTICA_CANVI_FORMAT" />
    <xs:enumeration value="COPIA_ELECT_AUTENTICA_PAPER"
/>
    <xs:enumeration value="COPIA_ELECT_AUTENTICA_PARCIAL"
/>
    <xs:enumeration value="ALTRES" />
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="ntiOrigen">
  <xs:restriction base="xs:string">

```



```

        <xs:enumeration value="CIUTADA" />
        <xs:enumeration value="ADMINISTRACIO" />
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="ntiTipoDocumento">
    <xs:restriction base="xs:string">
        <xs:enumeration value="RESOLUCIO" />
        <xs:enumeration value="ACORD" />
        <xs:enumeration value="CONTRACTE" />
        <xs:enumeration value="CONVENI" />
        <xs:enumeration value="DECLARACIO" />
        <xs:enumeration value="COMUNICACIO" />
        <xs:enumeration value="NOTIFICACIO" />
        <xs:enumeration value="PUBLICACIO" />
        <xs:enumeration value="JUSTIFICANT_RECEPCIO" />
        <xs:enumeration value="ACTA" />
        <xs:enumeration value="CERTIFICAT" />
        <xs:enumeration value="DILIGENCIA" />
        <xs:enumeration value="INFORME" />
        <xs:enumeration value="SOLICITUD" />
        <xs:enumeration value="DENUNCIA" />
        <xs:enumeration value="ALEGACIO" />
        <xs:enumeration value="RECURS" />
        <xs:enumeration value="COMUNICACIO_CIUTADA" />
        <xs:enumeration value="FACTURA" />
        <xs:enumeration value="ALTRES_INCAUTATS" />
        <xs:enumeration value="ALTRES" />
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="sicresTipoDocumento">
    <xs:restriction base="xs:string">
        <xs:enumeration value="FORMULARI" />
        <xs:enumeration value="ADJUNT" />
        <xs:enumeration value="TECNIC_INTERN" />
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="sicresValidezDocumento">
    <xs:restriction base="xs:string">
        <xs:enumeration value="COPIA" />
        <xs:enumeration value="COPIA_ELEC_AUTENTICA" />
        <xs:enumeration value="ORIGINAL" />
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="documentTipus">
    <xs:restriction base="xs:string">
        <xs:enumeration value="NIF" />
        <xs:enumeration value="CIF" />
        <xs:enumeration value="PASSAPORT" />
        <xs:enumeration value="NIE" />
        <xs:enumeration value="CODI_ORIGEN" />
        <xs:enumeration value="ALTRES" />
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="interessatTipus">
    <xs:restriction base="xs:string">
        <xs:enumeration value="PERSONA_FISICA" />
        <xs:enumeration value="PERSONA_JURIDICA" />
        <xs:enumeration value="ADMINISTRACIO" />
    </xs:restriction>
</xs:simpleType>

```

```

        </xs:restriction>
      </xs:simpleType>
      <xs:simpleType name="estat">
        <xs:restriction base="xs:string">
          <xs:enumeration value="PENDENT" />
          <xs:enumeration value="REBUDA" />
          <xs:enumeration value="PROCESSADA" />
          <xs:enumeration value="REBUTJADA" />
          <xs:enumeration value="ERROR" />
        </xs:restriction>
      </xs:simpleType>
    </xs:schema>
  </wsdl:types>
  <wsdl:message name="canviEstatResponse">
    <wsdl:part element="tns:canviEstatResponse" name="parameters">
    </wsdl:part>
  </wsdl:message>
  <wsdl:message name="consulta">
    <wsdl:part element="tns:consulta" name="parameters">
    </wsdl:part>
  </wsdl:message>
  <wsdl:message name="consultaResponse">
    <wsdl:part element="tns:consultaResponse" name="parameters">
    </wsdl:part>
  </wsdl:message>
  <wsdl:message name="canviEstat">
    <wsdl:part element="tns:canviEstat" name="parameters">
    </wsdl:part>
  </wsdl:message>
  <wsdl:portType name="BackofficeIntegracio">
    <wsdl:operation name="consulta">
      <wsdl:input message="tns:consulta" name="consulta">
      </wsdl:input>
      <wsdl:output message="tns:consultaResponse"
name="consultaResponse">
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="canviEstat">
      <wsdl:input message="tns:canviEstat" name="canviEstat">
      </wsdl:input>
      <wsdl:output message="tns:canviEstatResponse"
name="canviEstatResponse">
      </wsdl:output>
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="BackofficeIntegracioServiceSoapBinding"
type="tns:BackofficeIntegracio">
    <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="consulta">
      <soap:operation soapAction="" style="document" />
      <wsdl:input name="consulta">
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output name="consultaResponse">
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>

```

```
<wsdl:operation name="canviEstat">
  <soap:operation soapAction="" style="document" />
  <wsdl:input name="canviEstat">
    <soap:body use="literal" />
  </wsdl:input>
  <wsdl:output name="canviEstatResponse">
    <soap:body use="literal" />
  </wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:service name="BackofficeIntegracioService">
  <wsdl:port binding="tns:BackofficeIntegracioServiceSoapBinding"
    name="BackofficeIntegracioServicePort">
    <soap:address
      location="http://PROGRAMACI019:8081/distribucio/ws/backofficeIntegracio" />
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>
```

Annex IV: Exemple d'ús de la llibreria de *distribucio-backoffice-utils*

```
package es.caib.distribucio.core.service.ws.backoffice;

import java.util.Arrays;
import java.util.Date;
import java.util.List;
import java.util.Map;
import java.util.Properties;

import javax.xml.ws.WebService;

import org.apache.commons.lang.builder.ToStringBuilder;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.stereotype.Component;

import es.caib.distribucio.backoffice.utils.arxiu.ArxiuPluginListener;
import es.caib.distribucio.backoffice.utils.arxiu.ArxiuResultat;
import es.caib.distribucio.backoffice.utils.arxiu.ArxiuResultatAnnex;
import es.caib.distribucio.backoffice.utils.arxiu.BackofficeArxiuUtils;
import es.caib.distribucio.backoffice.utils.arxiu.BackofficeArxiuUtilsImpl;
import es.caib.distribucio.backoffice.utils.arxiu.DistribucioArxiuError;
import es.caib.distribucio.backoffice.utils.sistra.BackofficeSistra2Utils;
import es.caib.distribucio.backoffice.utils.sistra.BackofficeSistra2UtilsImpl;
import es.caib.distribucio.backoffice.utils.sistra.formulario.Formulario;
import es.caib.distribucio.backoffice.utils.sistra.pago.Pago;
import es.caib.distribucio.core.api.dto.ExpedientEstatEnumDto;
import es.caib.distribucio.core.api.service.ws.backoffice.AnotacioRegistreId;
import es.caib.distribucio.core.api.service.ws.backoffice.BackofficeWsService;
import es.caib.distribucio.core.helper.IntegracioHelper;
import es.caib.distribucio.core.helper.PropertiesHelper;
import es.caib.distribucio.core.helper.RegistreHelper;
import es.caib.distribucio.plugin.SistemaExternException;
import es.caib.distribucio.ws.backofficeintegracio.Annex;
import es.caib.distribucio.ws.backofficeintegracio.AnotacioRegistreEntrada;
import es.caib.distribucio.ws.backofficeintegracio.BackofficeIntegracio;
import es.caib.distribucio.ws.client.BackofficeIntegracioWsClientFactory;
import es.caib.plugins.arxiu.api.ContingutArxiu;
import es.caib.plugins.arxiu.api.Document;
import es.caib.plugins.arxiu.api.Expedient;
import es.caib.plugins.arxiu.api.IArxiuPlugin;

/**
 * Exemple de WS Backoffice Distribucio per rebre peticions d'anotacions pendents
 *
 * @author Limit Technologies <limit@limit.es>
 */
@Component
@WebService(
    name = "Backoffice",
    serviceName = "BackofficeService",
    portName = "BackofficeServicePort",
    endpointInterface = "es.caib.ripea.core.api.service.ws.BackofficeWsServiceBean",
    targetNamespace = "http://www.caib.es/distribucio/ws/backoffice")
public class BackofficeWsServiceImpl implements BackofficeWsService,

    ArxiuPluginListener {

    /** Instància del plugin d'Arxiu */
    private IArxiuPlugin arxiuPlugin = null;

    /** Mètode del WS que rep les comunicacions d'anotacions pendents. */
    @Override
    public void comunicarAnotacionsPendents(List<AnotacioRegistreId> ids) {

        try {
```

```
// Client dels serveis web de backoffice per consultar anotacions
BackofficeIntegracio backofficeClient = getBackofficeIntegracioServicePort();
es.caib.distribucio.ws.backofficeintegracio.AnotacioRegistreId idWs;
for (AnotacioRegistreId id : ids) {
    try {
        // Construeix l'identificador pel WS del backoffice de DISTRIBUCIO
        idWs = new
es.caib.distribucio.ws.backofficeintegracio.AnotacioRegistreId();
        idWs.setClauAcces(id.getClauAcces());
        idWs.setIndetificador(id.getIndetificador());

        // Consulta l'anotació
        AnotacioRegistreEntrada anotacio = backofficeClient.consulta(idWs);
        // Canvia l'estat a Rebuda
        backofficeClient.canviEstat(
            idWs,

es.caib.distribucio.ws.backofficeintegracio.Estat.REBUDA,
            "Canviar l'estat a rebuda");

        // Prepara la cria a la llibreria d'utilitats pel backoffice de
DISTRIBUCIO

        // Constructor amb la referència al plugin d'Arxiu
        BackofficeArxiuUtils backofficeArxiuUtils = new
BackofficeArxiuUtilsImpl(getArxiuPlugin());
        // Afegeix la instància de la classe com a escoltador d'events
        backofficeArxiuUtils.setArxiuPluginListener(this);
        // Estableix la carpeta on guardar els annexos de l'anotació
        backofficeArxiuUtils.setCarpeta(anotacio.getIdentificador());

        // Crida a la creació de l'expedient
        String SERIE_DOCUMENTAL = "S0002";
        String CLASSIFICACIO = "000000";
        ArxiuResultat arxiuResultat;
        int intent = 0;
        String expedientUuid = null;
        do {
            // Crida al mètode de creació de la llibreria
            arxiuResultat =
backofficeArxiuUtils.crearExpedientAmbAnotacioRegistre(
                expedientUuid,
                anotacio.getIdentificador(),
                null,
                Arrays.asList("A04019281"),
                new Date(),
                CLASSIFICACIO,
                ExpedientEstatEnumDto.OBERT,
                SERIE_DOCUMENTAL,
                anotacio);
            expedientUuid = arxiuResultat.getIdentificadorExpedient();
            // Imprimeix el resultat per pantalla
            this.logResultat(
                "Resultat de la crida " + intent++ + "
crearExpedientAmbAnotacioRegistre per l'expedient" + anotacio.getIdentificador(),
                arxiuResultat);
        } while(arxiuResultat.getErrorCodi() !=
DistribucioArxiuError.NO_ERROR && intent < 10);

        // Consultem el nou expedient a l'Arxiu
        if (expedientUuid != null) {
            Expedient expedientDetalls =
getArxiuPlugin().expedientDetalls(arxiuResultat.getIdentificadorExpedient(), null);
            logger.debug("S'ha creat l'expedient \" " +
expedientDetalls.getNom() + "\" amb id=" + expedientDetalls.getIdentificador() + " per l'anotació " +
anotacio.getIdentificador() + " amb els següents continguts:");
            for (ContingutArxiu contingut :
expedientDetalls.getContinguts()) {
                logger.debug("- " + contingut.getIdentificador() +
" " + contingut.getNom() + " amb " + (contingut.getFirmes() != null? contingut.getFirmes().size() : 0) + "
firmes.");
            }
        } else {
            logger.warn("L'expedient no s'ha pogut crear per l'anotació
```

```
" + anotacio.getIdentificador());

    }
    // Processament dels annexos de documents tècnics segons el títol
    String titol;
    for (Annex annex : anotacio.getAnnexos()) {
        titol = annex.getTitol();
        if (titol != null
            && ("FORMULARIO".equals(titol)
                || "PAGO".equals(titol)))
        {
            // Recupera el contingut de l'annex
            Document document =
            getArxiuPlugin().documentDetalls(annex.getUuid(), null, true);
            byte[] contingut =
            document.getContingut().getContingut();
            // Interpreta el contingut amb la classe
            BackofficeSistra2Utils
            BackofficeSistra2UtilsImpl();
            Logger.debug(" Document tècnic \" + titol + \"\n.
            Dades:");

            try {
                if ("FORMULARIO".equals(titol)) {
                    Formulario formulario =
                    Logger.debug(" formulario: " +
                } else if ("PAGO".equals(titol)) {
                    Pago pago =
                    Logger.debug(" pago: " +
                }
            } catch (Exception e) {
                Logger.error("Error obtenint la informació
            del document tècnic " + titol + ":" + e.getMessage(), e);
        }
    }

    // Es comunica el resultat a DISTRIBUCIO
    switch(arxiuResultat.getErrorCodi()) {
        case 0:
            backofficeClient.canviEstat(idWs,
            es.caib.distribucio.ws.backofficeintegracio.Estat.PROCESSADA, "Processada");
            break;
        default:
            backofficeClient.canviEstat(idWs,
            es.caib.distribucio.ws.backofficeintegracio.Estat.ERROR, arxiuResultat.getErrorCodi() + " " +
            arxiuResultat.getErrorMessage());
            break;
    }
    } catch (Throwable ex) {
        Logger.error("Error al processant la petició d'anotació amb id " +
        id.getIdetificador(), ex);
    }
    } catch (Exception e) {
        Logger.error("Error no controlat en el mètode de comunicació d'anotacions pendents:
        " + e.getMessage(), e);
    }
}

private void logResultat(String descripcio, ArxiuResultat arxiuResultat) {
    Logger.debug(descripcio);
    // Resultat a nivell d'expedient
    Logger.debug("- uuid: " + arxiuResultat.getIdentificadorExpedient());
    Logger.debug("- accio: " + arxiuResultat.getAccio());
    Logger.debug("- errorCodi: " + arxiuResultat.getErrorCodi());
}
```

```

        logger.debug("- errorMessage: " + arxiuResultat.getErrorMessage());
        logger.debug("- excepcio: " + (arxiuResultat.getException() != null?
arxiuResultat.getException().getClass() + " " + arxiuResultat.getException().getMessage() : " - "));

// Resultat pels annexos
List<ArxiuResultatAnnex> resultatAnnexos = arxiuResultat.getResultatAnnexos();
logger.debug(" - Resultat dels " + resultatAnnexos.size() + " annexos:");
for (ArxiuResultatAnnex resultatAnnex : resultatAnnexos) {
    logger.debug("\b- uuid: " + resultatAnnex.getIdentificadorAnnex());
    logger.debug("\b- accio: " + resultatAnnex.getAccio());
    logger.debug("\b- errorCodi: " + resultatAnnex.getErrorCodi());
    logger.debug("\b- errorMessage: " + resultatAnnex.getErrorMessage());
    logger.debug("\b- excepcio: " + (resultatAnnex.getException() != null?
resultatAnnex.getException().getClass() + " " + resultatAnnex.getException().getMessage() : " - "));
}
}

private IArxiuPlugin getArxiuPlugin() throws SistemaExternException {
    if (arxiuPlugin == null) {
        String pluginClass = getPropertyPluginArxiu();
        if (pluginClass != null && pluginClass.length() > 0) {
            try {
                Class<?> clazz = Class.forName(pluginClass);
                if (PropertiesHelper.getProperties().isLlegirSystem()) {
                    arxiuPlugin = (IArxiuPlugin)clazz.getDeclaredConstructor(
                        String.class).newInstance(
                            "es.caib.distribucio.");
                } else {
                    arxiuPlugin = (IArxiuPlugin)clazz.getDeclaredConstructor(
                        String.class,
                        Properties.class).newInstance(
                            "es.caib.distribucio.",
                            PropertiesHelper.getProperties().findAll());
                }
            } catch (Exception ex) {
                throw new SistemaExternException(
                    IntegracioHelper.INTCODI_ARXIU,
                    "Error al crear la instància del plugin d'arxiu
digital",
                    ex);
            }
        } else {
            throw new SistemaExternException(
                IntegracioHelper.INTCODI_ARXIU,
                "No està configurada la classe per al plugin d'arxiu
digital");
        }
    }
    return arxiuPlugin;
}

private String getPropertyPluginArxiu() {
    return PropertiesHelper.getProperties().getProperty(
        "es.caib.distribucio.plugin.arxiu.class");
}

private BackofficeIntegracio getBackofficeIntegracioServicePort() throws Exception {
    BackofficeIntegracio wsClient = null;
    String url =
PropertiesHelper.getProperties().getProperty("es.caib.distribucio.backoffice.test.backofficeIntegracio.url");
;
    String usuari =
PropertiesHelper.getProperties().getProperty("es.caib.distribucio.backoffice.test.backofficeIntegracio.usuar
i");
    String contrasenya =
PropertiesHelper.getProperties().getProperty("es.caib.distribucio.backoffice.test.backofficeIntegracio.contr
asenya");
    if (url != null && usuari != null && contrasenya != null) {
        logger.debug(">>> Creant el client BackofficeIntegracio WS");
        wsClient = BackofficeIntegracioWsClientFactory.getWsClient(
            url,
            usuari,
            contrasenya);
    }
}

```



```

    } else {
        throw new RuntimeException("Falta configurar les propietats pel client de
Backoffice de DISTRIBUCIO es.caib.distribucio.backoffice.test.backofficeIntegracio.*");
    }
    return wsClient;
}

private static final Logger logger = LoggerFactory.getLogger(RegistreHelper.class);

/** Mètode que la llibreria del backoffice de distribució crida si es fixa una instància que
implementa {@link ArxiuPluginListener}.
 *
 */
@Override
public void event(
    String metode,
    Map<String, String> parametres,
    boolean correcte,
    String error,
    Exception e,
    long timeMs) {
    StringBuilder str = new StringBuilder()
        .append("S'ha invocat el mètode \"")
        .append(metode)
        .append("\" amb els paràmetres {");

    int i = 0;
    for (String key : parametres.keySet()) {
        str.append(key).append("=").append(parametres.get(key));
        i++;
        if (i < parametres.size())
            str.append(", ");
    }
    str.append("} amb resultat ").append(correcte ? "OK" : "KO");
    if (error != null)
        str.append(": ").append(error);
    if (e != null)
        str.append(" ").append(e.getClass()).append(" ").append(e.getMessage());
    str.append(" ").append(timeMs).append("ms");

    logger.debug(str.toString());
}
}

```