



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

API REST ROLSAC : Manual d'integració

15 de Novembre de 2019

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

Lot 2 (Serveis electrònics per a la ciutadania)

Oficina Tècnica de Direcció de Projecte

Control de versions del document

Control de Canvis			
Data	Autor	Versió	Canvis
19/06/2019	Indra	v1.0	Versió inicial
15/11/2019	Indra	v1.1	Actualización

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.....	4
2. Diagrama de funcionament.....	4
3. Diagrama de classes.....	7
4. Descripció dels serveis.....	9
5. Consideracions de disseny i ús de l'API.....	9
5.1. <i>Endpoints</i> dels serveis.....	9
5.2. Segurització dels serveis.....	9
6. Client d'exemple.....	10
6.1. Dependències Maven.....	10
6.2. Classes.....	10
6.2.1. Fitxer Rlink.java.....	11
6.2.2. Fitxer RTramiteRolsac.java.....	11
6.2.3. Fitxer RTasaRolsac.java.....	14
6.2.4. Fitxer RRespuestaBasicaRolsac.java.....	15
6.2.5. Fitxer RRespuestaTramites.java.....	16
6.2.6. Fitxer RolsacRest.java – Cliente REST.....	16

1. Objecte

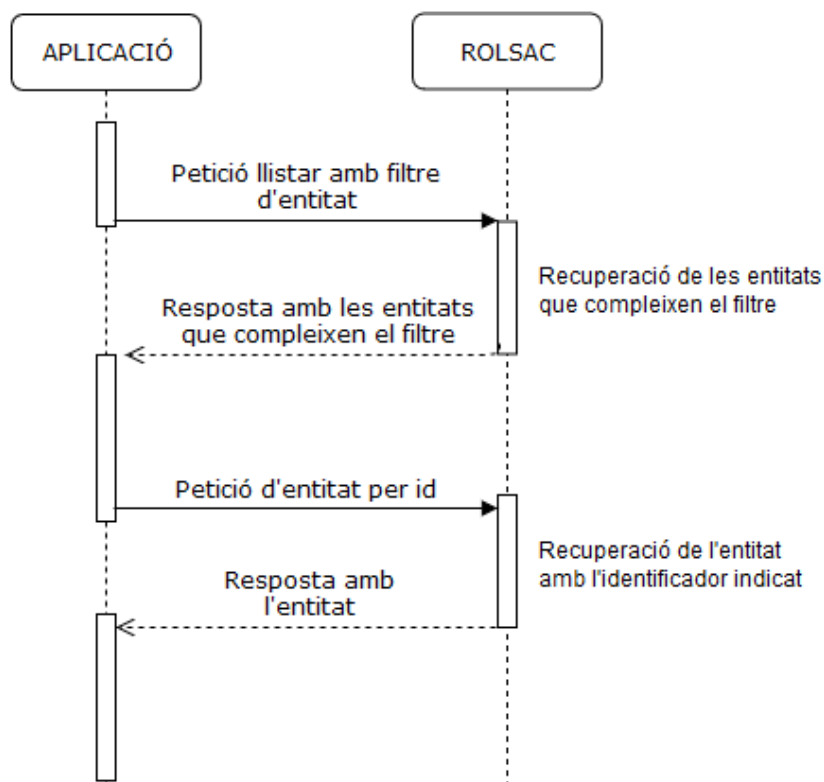
En aquest document s'aporta tota la informació necessària perquè qualsevol aplicació es pugui integrar amb ROLSAC mitjançant l'API REST, y d'aquesta manera consultar la informació emmagatzemada a ROLSAC .

2. Diagrama de funcionament

L'API REST publicada a ROLSAC permet consultar la informació emmagatzemada a l'aplicació, per tant totes les operacions publicades a ROLSAC son de consulta, amb la excepció de les funcions d'estadístiques, que permeten augmentar un nombre intern amb els accessos a UA's, fitxes, normatives i procediments. Bàsicament existeixen 2 tipus de funcions a l'API per a cada entitat:

- Funcions per recuperar un llistat d'elements en funció de determinats filtres.
- Funcions per recuperar un element en funció del seu identificador.

A continuació es mostra el diagrama de funcionament habitual, en el que es realitzen 2 peticions a l'api, una per recuperar un llistat d'entitats i una altra per recuperar un element en funció del seu identificador.



Per a permetre enviar informació i filtrar els resultats es farà ús dels paràmetres d'entrada, els quals poden ser de 5 tipus:

- **Inclosos a la URL** : Paràmetres obligatoris, que han de ser inclosos per a obtenir resultats. Per exemple, el codi d'un element, si intentem obtenir una Normativa concreta: {urlBase}/Normatives/123
- **Paràmetres de Paginació**: Paràmetres opcionals que permeten configurar la paginació. Aquest objecte haurà de venir serialitzat en JSON i inclourà els paràmetres "page" i "size". Aquests paràmetres tindran un valor per defecte per al cas que no vengen informats. D'aquesta manera es permet fitar les cerques, encara que no vengen informats.
- **Paràmetre d'Idioma**: Paràmetre opcional que permet configurar l'idioma. Aquest paràmetre simple tindrà un valor per defecte per al cas que no vingui informat.
- **Paràmetres de filtre**: Paràmetres Opcionals que permeten fitar les cerques, adaptant-se a cada entitat específica. A priori aquests paràmetres són els filtres específics de cada entitat, que seran contemplats com a objectes independents, serialitzats en JSON.
- **Paràmetres d'ordenació** : Aquests paràmetres permetran ordenar les cerques pels camps indicats, en manera ascendent o descendent. Aquest objecte serà serialitzat en JSON.

Tots aquests paràmetres a excepció dels inclosos a l'url serán enviats per POST.

Les respostes de totes les funcions publicades sempre tenen la mateixa estructura, permetent l'homogeneïtat de l'aplicació. Totes les funcions retornen un objecte JSON en el qual s'inclou com a mínim els següents camps:

- **Status**: indica el resultado de la operación.
- **Missatge d'Error**: Descripció de l'error. Únicament informat si l'aplicació retorna un estat diferent de 200
- **Nombre total d'elements** : indica el nombre total d'elements coincidents amb la cerca
- **Resultat**: conté un objecte amb el resultat de la cerca. Pot contenir un objecte simple o un llistat d'objectes. El contingut d'aquest camp varia en funció de l'entitat que retorna, es pot consultar el detall per a cada funció al swagger.

A continuació, es mostra, a mode d'exemple, les possibles respostes retornada per l'api:

Resposta OK

```
{
  "status": "200",
  "numElementos": 1,
  "resultado": {
    "campo1": "valor1",
    "campo2": "valor2",
    "campo3": "valor3"
  }
}
```

Resposta ERROR

```
{
  "status": "400",
  "MensajeError": "Petición incorrecta"
  "numElementos": 0,
  "resultado": ""
}
```

Resposta Simple OK

```
{
  "status": "200",
  "numElementos": 1,
  "resultado": "Texte"
}
```

En les entitats en les quals s'indiqui un valor relacionat amb una altra entitat, s'afegirà un camp `link_entitat` que contindrà la informació de la relació. A tall d'exemple en l'entitat "Document" s'afegiran uns camps `link_archivo` i `link_ficha` que contindran l'objecte de relació amb les entitats "Arxiu", "Fitxa" respectivament, a continuació, es mostra un resultat d'exemple per a aquesta entitat:

```
{
  "codigo": "595152",
  "idioma": "ca",
  "link_archivo": {
    "codigo": "595153",
    "href": "http://host:port/rolsac/api/rest/v1/archivos/595153",
    "rel": "archivos"
  },
  "link_ficha": {
    "codigo": "595142",
    "href": "http://host:port/api/rest/v1/fichas/595142",
    "rel": "fichas"
  },
  "orden": "0",
  "titulo": "titulo del documento"
}
```

3. Diagrama de classes

Donat que aquesta API està documentada mitjançant Swagger, es pot consultar el detall del camp de totes les entitats relacionades a la URL `http://host:port/rolsac/api/rest`, on *host* i *port* hauràn de ser substituïts per les dades de l'entorn emprat.

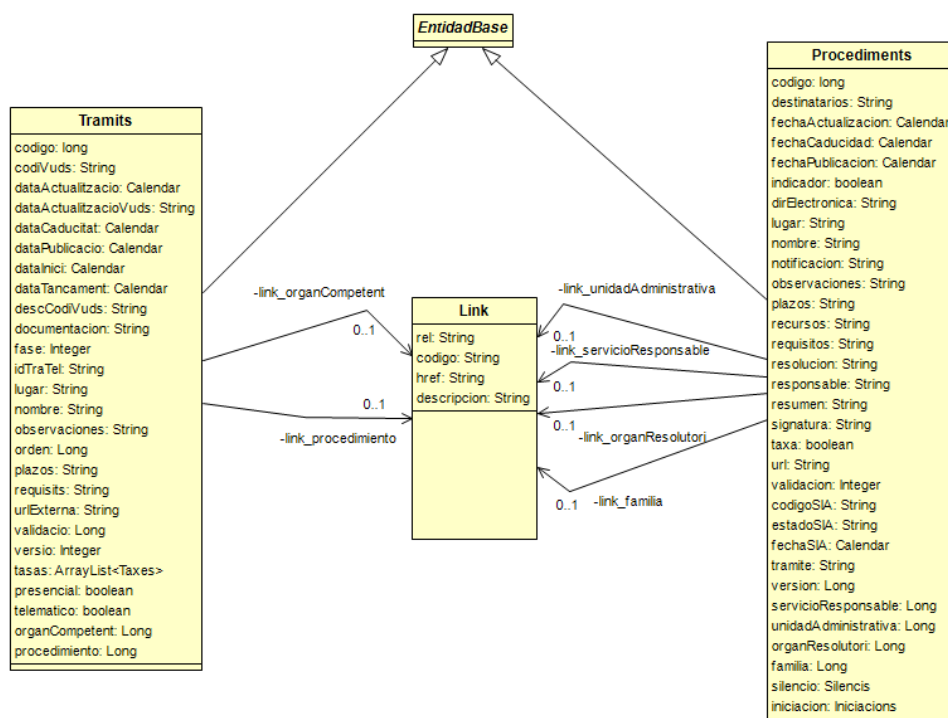
En cas de voler modelitzar mitjançant JAVA l'accés a l'api, Es pot emprar com a ajuda les classes definides a l'aplicació de ROLSAC dins la url de github

<https://github.com/GovernIB/rolsac/tree/rolsac-1.8/moduls/apirest/src/es/caib/rolsac/apirest/v1/model>

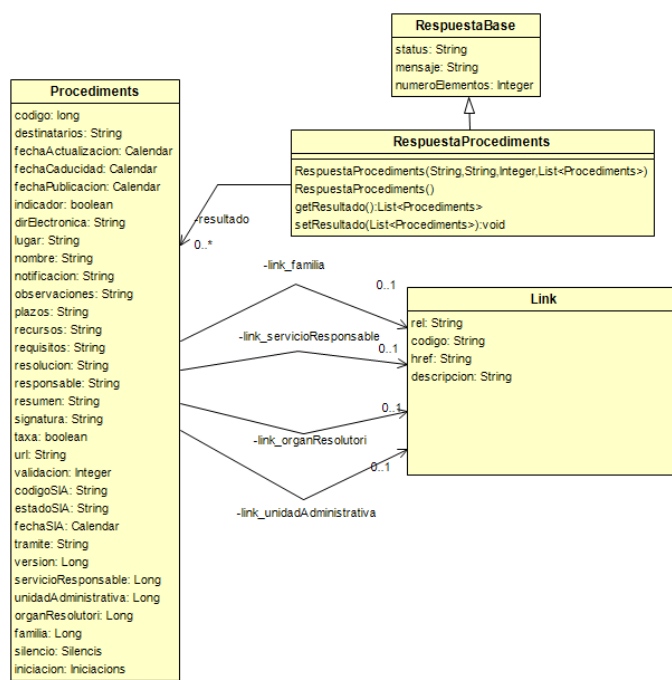
- ▼ model
 - > filtros
 - > orden
 - > respuestas
 - AgrupacioFetVital.java
 - AgrupacioMateries.java
 - Anxius.java
 - Bulletins.java
 - CatalegDocuments.java
 - Documents.java
 - DocumentsNormatives.java
 - DocumentsTramits.java
 - Edificis.java
 - Enllacos.java
 - EntidadBase.java
 - EspaisTerritorials.java
 - excepcioDocumentacio.java
 - Families.java
 - FetsVitals.java
 - Fitxes.java
 - FitxesUA.java
 - Formularis.java
 - IconesFamilies.java
 - Idioma.java
 - Iniciacions.java
 - Link.java
 - MateriaAgrupacio.java
 - Materies.java
 - Normatives.java
 - Perfil.java
 - Personal.java
 - Procediments.java
 - PublicsObjectius.java
 - Seccions.java
 - Serveis.java
 - Silencis.java
 - Taxes.java
 - Tipus.java
 - TipusAfectacio.java
 - Tramits.java
 - Tratamiento.java
 - UnitatAdministrativa.java
 - UnitatsMateries.java
 - Usuaris.java

Dintre la carpeta model, trobam les classes que modelitzen les entitats que s'utilitzen dins l'api. També es poden consultar els diferents filtres, filtres d'ordenació i respostes que retorna l'api.

A mode d'exemple tal com s'ha indicat anteriorment les entitats relacionades amb altres es referencien per la classe link, en aquest cas un tràmit pertany a un determinat procediment, per tant, un dels seus atributs serà link_procedimiento de tipus link, i d'aquesta manera es podrà consultar el codi del procediment(codigo), a quina entitat fa referencia(rel) i la url d'accés al recurs(href), a continuació es mostra una imatge que reflexa aquesta estructura.



Per altra banda totes les repòses de l'api segueixen la mateixa estructura, heretant totes de RespuestaBase, per exemple, la resposta dels procediments segueix el següent model de classes:



4. Descripció dels serveis

Tota la informació relativa al serveis de l'API, paràmetres d'entrada i valors de retorn es pot trobar a la pàgina de Swagger (<http://host:port/rolsac/api/rest/>).

5. Consideracions de disseny i ús de l'API

5.1. Endpoints dels serveis

Tots els *endpoints* dels diferents serveis de l'API de ROLSAC es poden trobar a la pàgina de Swagger <http://host:port/rolsac/api/rest/>. Es necessari revisar el valor de «host» i «port» en funció de l'entorn al que s'hi vulgui accedir.

A mode de resum, per cada entitat de ROLSAC, Existeixen 2 tipus de funcions, les quals s'accedeixen pel seu *endpoint* particular:

- Les funcions per llistar entitats: <http://host:port/rolsac/api/rest/v1/{entitat}>
- Les funcions per obtenir un element en funció del seu identificador: <http://host:port/rolsac/api/rest/v1/{entitat}/{codi}>

on {entitat} es el nom de l'entitat a cercar, i {codi} es l'identificador de l'entitat.

5.2. Segurització dels serveis

Els serveis de l'aplicació estan securitzats amb autenticació, BASIC amb usuari/contrasenya.

L'usuari emprar per a consumir els serveis haurà de tenir assignat el rol RSC_API.

6. Client d'exemple

No existeix un client de referencia pel consum dels serveis de l'API de ROLSAC, però a continuació es presenta, a mode d'exemple, una serie de classes que implementen un client per emprar les funcions publicades a l'api. En aquest cas, es permet la consulta de tràmits mitjançant l'obtenció d'un JSON i mitjançant la modelització de la classe de tràmits.

6.1. Dependències Maven

Per a desenvolupar aquest client per consultar l'API REST de ROLSAC el més senzill és crear un nou projecte amb Maven i incloure els següents blocs a dins el fitxer pom.xml:

A dins la secció <dependencies>:

```

<dependencies>
  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-web</artifactId>
    <version>4.3.13.RELEASE</version>
  </dependency>
  <dependency>
    <groupId>com.fasterxml.jackson.core</groupId>
    <artifactId>jackson-databind</artifactId>
    <version>2.8.10</version>
  </dependency>
</dependencies>

```

6.2. Classes

A continuació es mostren les classes que formen part del client, essent RolsacRest la classe principal:

```

src
├── main
│   └── java
│       ├── RLink
│       ├── RolsacRest
│       ├── RRespostaBasicaRolsac
│       ├── RRespostaTramites
│       ├── RTasaRolsac
│       └── RTramiteRolsac

```

6.2.1. Fitxer RLink.java

```

public class RLink {
    public RLink(final String entidad, final String codigo, final String url,
        final String descripcion, final boolean hateoas) {
        super();
        this.rel = entidad;
        this.codigo = codigo;
        this.descripcion = descripcion;
        if (hateoas) {
            this.href = url;
        } else {
            this.href = null;
        }
    }

    public RLink(final String entidad, final String codigo, final String url,
        final boolean hateoas) {
        this(entidad, codigo, url, null, hateoas);
    }

    public RLink() {
        super();
    }

    private String rel;
    private String codigo;
    private String href;
    private String descripcion;
    public String getRel() {
        return rel;
    }

    public void setRel(final String rel) {
        this.rel = rel;
    }

    public String getCodigo() {
        return codigo;
    }

    public void setCodigo(final String codigo) {
        this.codigo = codigo;
    }

    public String getHref() {
        return href;
    }

    public void setHref(final String href) {
        this.href = href;
    }

    public String getDescripcion() {
        return descripcion;
    }

    public void setDescripcion(final String descripcion) {
        this.descripcion = descripcion;
    }
}

```

6.2.2. Fitxer RTramiteRolsac.java

```

public class RTramiteRolsac {
    private long codigo;
    private String codiVuds;
    private java.util.Calendar dataActualitzacio;
    private String dataActualitzacioVuds;
    private java.util.Calendar dataCaducitat;
    private java.util.Calendar dataPublicacio;
    private java.util.Calendar dataInici;
    private java.util.Calendar dataTancament;
    private String descCodiVuds;
    private String documentacion;
    private Integer fase;
    private String idTraTel;
    private String lugar;
    private String nombre;
    private String observaciones;
    private Long orden;
    private String plazos;
}

```

```
private String requisits;
private String urlExterna;
private Long validacio;
private Integer versio;
private RTasaRolsac[] tasas;
private boolean presencial;
private boolean telematico;
private RLink link_organCompetent;
private RLink link_procedimiento;
private Long organCompetent;
private Long procedimiento;
public long getCodigo() {
    return codigo;
}
public void setCodigo(final long codigo) {
    this.codigo = codigo;
}
public String getCodiVuds() {
    return codiVuds;
}
public void setCodiVuds(final String codiVuds) {
    this.codiVuds = codiVuds;
}
public java.util.Calendar getDataActualitzacio() {
    return dataActualitzacio;
}
public void setDataActualitzacio(final java.util.Calendar dataActualitzacio) {
    this.dataActualitzacio = dataActualitzacio;
}
public String getDataActualitzacioVuds() {
    return dataActualitzacioVuds;
}
public void setDataActualitzacioVuds(final String dataActualitzacioVuds) {
    this.dataActualitzacioVuds = dataActualitzacioVuds;
}
public java.util.Calendar getDataCaducitat() {
    return dataCaducitat;
}
public void setDataCaducitat(final java.util.Calendar dataCaducitat) {
    this.dataCaducitat = dataCaducitat;
}
public java.util.Calendar getDataPublicacio() {
    return dataPublicacio;
}
public void setDataPublicacio(final java.util.Calendar dataPublicacio) {
    this.dataPublicacio = dataPublicacio;
}
public java.util.Calendar getDataInici() {
    return dataInici;
}
public void setDataInici(final java.util.Calendar dataInici) {
    this.dataInici = dataInici;
}
public java.util.Calendar getDataTancament() {
    return dataTancament;
}
public void setDataTancament(final java.util.Calendar dataTancament) {
    this.dataTancament = dataTancament;
}
public String getDescCodiVuds() {
    return descCodiVuds;
}
public void setDescCodiVuds(final String descCodiVuds) {
    this.descCodiVuds = descCodiVuds;
}
public String getDocumentacion() {
    return documentacion;
}
public void setDocumentacion(final String documentacion) {
    this.documentacion = documentacion;
}
public Integer getFase() {
    return fase;
}
public void setFase(final Integer fase) {
```

```

        this.fase = fase;
    }
    public String getIdTraTel() {
        return idTraTel;
    }
    public void setIdTraTel(final String idTraTel) {
        this.idTraTel = idTraTel;
    }
    public String getLugar() {
        return lugar;
    }
    public void setLugar(final String lugar) {
        this.lugar = lugar;
    }
    public String getNombre() {
        return nombre;
    }
    public void setNombre(final String nombre) {
        this.nombre = nombre;
    }
    public String getObservaciones() {
        return observaciones;
    }
    public void setObservaciones(final String observaciones) {
        this.observaciones = observaciones;
    }
    public Long getOrden() {
        return orden;
    }
    public void setOrden(final Long orden) {
        this.orden = orden;
    }
    public String getPlazos() {
        return plazos;
    }
    public void setPlazos(final String plazos) {
        this.plazos = plazos;
    }
    public String getRequisits() {
        return requisits;
    }
    public void setRequisits(final String requisits) {
        this.requisits = requisits;
    }
    public String getUrlExterna() {
        return urlExterna;
    }
    public void setUrlExterna(final String urlExterna) {
        this.urlExterna = urlExterna;
    }
    public Long getValidacio() {
        return validacio;
    }
    public void setValidacio(final Long validacio) {
        this.validacio = validacio;
    }
    public Integer getVersio() {
        return versio;
    }
    public void setVersio(final Integer versio) {
        this.versio = versio;
    }
    public RTasaRolsac[] getTasas() {
        return tasas;
    }
    public void setTasas(final RTasaRolsac[] tasas) {
        this.tasas = tasas;
    }
    public RLink getLink_organCompetent() {
        return link_organCompetent;
    }
    public void setLink_organCompetent(final RLink link_organCompetent) {
        this.link_organCompetent = link_organCompetent;
    }
    public RLink getLink_procedimiento() {

```

```

        return link_procedimiento;
    }
    public void setLink_procedimiento(final RLink link_procedimiento) {
        this.link_procedimiento = link_procedimiento;
    }
    public boolean isPresencial() {
        return presencial;
    }
    public void setPresencial(final boolean presencial) {
        this.presencial = presencial;
    }
    public boolean isTelematico() {
        return telematico;
    }
    public void setTelematico(final boolean telematico) {
        this.telematico = telematico;
    }
    public Long getOrganCompetent() {
        return organCompetent;
    }
    public void setOrganCompetent(final Long organCompetent) {
        this.organCompetent = organCompetent;
    }
    public Long getProcedimiento() {
        return procedimiento;
    }
    public void setProcedimiento(final Long procedimiento) {
        this.procedimiento = procedimiento;
    }
}

```

6.2.3. Fitxer RTasaRolsac.java

```

public class RTasaRolsac {
    private long codigo;
    private java.lang.String codificacio;
    private java.lang.String descripcio;
    private java.lang.String formaPagament;
    private java.lang.Long orden;

    public RTasaRolsac() {
        super();
    }

    public void setId(Long codigo) {
        this.codigo = codigo;
    }

    public long getCodigo() {
        return codigo;
    }

    public void setCodigo(long codigo) {
        this.codigo = codigo;
    }

    public java.lang.String getCodificacio() {
        return codificacio;
    }

    public void setCodificacio(java.lang.String codificacio) {
        this.codificacio = codificacio;
    }

    public java.lang.String getDescripcio() {

```

```

    return descriptio;
  }

  public void setDescriptio(java.lang.String descriptio) {
    this.descriptio = descriptio;
  }

  public java.lang.String getFormaPagament() {
    return formaPagament;
  }

  public void setFormaPagament(java.lang.String formaPagament) {
    this.formaPagament = formaPagament;
  }

  public java.lang.Long getOrden() {
    return orden;
  }

  public void setOrden(java.lang.Long orden) {
    this.orden = orden;
  }
}

```

6.2.4. Fitxer RRespuestaBasicaRolsac.java

```

import com.fasterxml.jackson.core.type.TypeReference;
import com.fasterxml.jackson.databind.ObjectMapper;

import java.io.IOException;

public class RRespuestaBasicaRolsac {
  private String status;
  private String mensaje;
  private Integer numeroElementos;
  public RRespuestaBasicaRolsac(final String status, final String mensaje,
    final Integer numeroElementos) {
    super();
    this.status = status;
    this.mensaje = mensaje;
    this.numeroElementos = numeroElementos;
  }
  public RRespuestaBasicaRolsac() {
    this.status = null;
    this.mensaje = null;
    this.numeroElementos = null;
  }
  public String getStatus() {
    return status;
  }
  public void setStatus(final String status) {
    this.status = status;
  }
  public String getMensaje() {
    return mensaje;
  }
  public void setMensaje(final String mensaje) {
    this.mensaje = mensaje;
  }
  public Integer getNumeroElementos() {
    return numeroElementos;
  }
  public void setNumeroElementos(final Integer numeroElementos) {
    this.numeroElementos = numeroElementos;
  }

  public static RRespuestaBasicaRolsac valueOf(final String json) {

```

```

    final ObjectMapper objectMapper = new ObjectMapper();
    final TypeReference<RRespuestaBasicaRolsac> typeRef = new
TypeReference<RRespuestaBasicaRolsac>() {};
    RRespuestaBasicaRolsac obj;
    try {
        obj = (RRespuestaBasicaRolsac) objectMapper.readValue(json, typeRef);
    } catch (final IOException e) {
        throw new RuntimeException(e);
    }
    return obj;
}

```

6.2.5. Fitxer RRespuestaTramites.java

```

public class RRespuestaTramites extends RRespuestaBasicaRolsac {
    private RTramiteRolsac[] resultado;
    public RRespuestaTramites(final String status, final String mensaje, final Integer
numeroElementos,
        final RTramiteRolsac[] resultado) {
        super(status, mensaje, numeroElementos);
        this.resultado = resultado;
    }
    public RRespuestaTramites() {
        super();
    }
    public RTramiteRolsac[] getResultado() {
        return resultado;
    }
    public void setResultado(final RTramiteRolsac[] resultado) {
        this.resultado = resultado;
    }
}

```

6.2.6. Fitxer RolsacRest.java – Cliente REST

```

import org.springframework.http.HttpEntity;
import org.springframework.http.HttpHeaders;
import org.springframework.util.LinkedMultiValueMap;
import org.springframework.util.MultiValueMap;
import org.springframework.http.MediaType;
import org.springframework.http.ResponseEntity;
import org.springframework.http.client.BasicAuthorizationInterceptor;
import org.springframework.web.client.HttpClientErrorException;
import org.springframework.web.client.RestTemplate;

import static java.lang.System.exit;
import static java.nio.charset.StandardCharsets.UTF_8;

public class RolsacRest {
    public RolsacRest() {
    }

    public static String endpoint = "http://host:puerto/rolsac/api/rest/v1";
    public static String entidad = "tramites";
    public static String idioma = "es";
    public static String filtro = "{}"; //json amb els filtres, consultar swagger per a mes
informació
    public static String filtroOrdenacion = "{}"; //json amb els filtres per ordenar els resultats,
consultar swagger per a mes informació

    public static String usr = "usuari";

```



```

public static String pw = "password";

public static void ejecutaListadoGenerico(String endpoint, String entidad, String filtro,
String filtroOrdenacion, String idioma) {
    System.out.println(" ");
    System.out.println("inicio ejecutaListadoGenerico");

    final RestTemplate restTemplate = new RestTemplate();
    restTemplate.getInterceptors().add(new BasicAuthorizationInterceptor(usr, pw));
    final HttpHeaders headers = new HttpHeaders();
    headers.setContentType(org.springframework.http.MediaType.APPLICATION_FORM_URLENCODED);

    // Obtener tramite.
    final MultiValueMap<String, String> map = new LinkedMultiValueMap<>();
    map.add("idioma", idioma);
    map.add("filtro", filtro);
    map.add("filtroOrdenacion", filtroOrdenacion);
    final HttpEntity<MultiValueMap<String, String>> request = new HttpEntity<>(map, headers);
    String res = "";
    try{
        final ResponseEntity<String> response = restTemplate.
            postForEntity(endpoint + "/" + entidad , request, String.class);
        res = response.getBody();
    }catch (HttpClientErrorException e){
        res = new String(e.getResponseBodyAsByteArray(), UTF_8);
    }
    System.out.println("Respuesta:" + res);
}

public static void ejecutaListadoTramites(String endpoint, String filtro, String
filtroOrdenacion, String idioma){
    System.out.println(" ");
    System.out.println("inicio ejecutaListadoTramites");

    final RestTemplate restTemplate = new RestTemplate();
    restTemplate.getInterceptors().add(new BasicAuthorizationInterceptor(usr, pw));
    final HttpHeaders headers = new HttpHeaders();
    headers.setContentType(MediaType.APPLICATION_FORM_URLENCODED);

    // Obtener tramite.
    final MultiValueMap<String, String> map = new LinkedMultiValueMap<>();
    map.add("idioma", idioma);
    map.add("filtro", filtro);
    map.add("filtroOrdenacion", filtroOrdenacion);
    final HttpEntity<MultiValueMap<String, String>> request = new HttpEntity<>(map, headers);
    try{
        final ResponseEntity<RRespuestaTramites> responseTramite = restTemplate
            .postForEntity(endpoint + "/tramites/" , request, RRespuestaTramites.class);

        if (responseTramite != null && responseTramite.getBody() != null){
            if(!responseTramite.getBody().getStatus().equals("200")){
                //error controlado
                System.out.println(responseTramite.getBody().getMensaje());
                System.out.println(responseTramite.getBody().getStatus());
                exit(0);
            }else if(responseTramite.getBody().getNumeroElementos().equals(0)){
                // Sin resultados
                System.out.println(responseTramite.getBody().getMensaje());
                System.out.println(responseTramite.getBody().getStatus());
                System.out.println(responseTramite.getBody().getNumeroElementos());
                exit(0);
            }else{
                // Hay resultados
                final RTramiteRolsac[] tramitesRolsac =
responseTramite.getBody().getResultado();

                if (tramitesRolsac == null || tramitesRolsac.length == 0) {
                    System.out.println("No hay tramite");
                    exit(0);
                }
                System.out.println("Se han encontrado "

```

```

        + responseTramite.getBody().getNumeroElementos()
        + " de los cuales se mostraran "
        + tramitesRolsac.length );
    // Obtener tramites.
    for (RTramiteRolsac tramiteRolsac:tramitesRolsac) {
        System.out.println("codigo procedimiento:" +
tramiteRolsac.getLink_procedimiento().getCodigo());
        System.out.println("Nombre:" + tramiteRolsac.getNombre());
        System.out.println(" ");
    }
}
} else {
    //error no controlado
    System.out.println("No se puede realizar la consulta ocurrio algún error");
}

} catch (HttpClientErrorException e) {
    RRespuestaBasicaRolsac res = RRespuestaBasicaRolsac.valueOf(new
String(e.getResponseBodyAsByteArray(), UTF_8));
    System.out.println(res.getMensaje());
    System.out.println(res.getStatus());
}
}

public static void main(String[] args) {
    ejecutaListadoGenerico( endpoint, entidad, filtro, filtroOrdenacion, idioma);
    ejecutaListadoTramites( endpoint, filtro, filtroOrdenacion, idioma);
}
}

```