



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

# PINBAL: Manual d'integració amb l'API REST

5 de maig de 2021

**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
29/04/2021	Límit Technologies	1.0	Versió inicial
25/02/2022	Límit Technologies	1.1	Afegit adreça de documentació REST

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

<b>Control de versions del document.....</b>	<b>2</b>
<b>1. Objecte.....</b>	<b>4</b>
<b>2. Serveis disponibles.....</b>	<b>4</b>
2.1. Serveis SCSP.....	4
2.1.1. Petició síncrona (peticionSincrona).....	4
2.1.2. Petició asíncrona (peticionAsincrona).....	5
2.1.3. Obtenir resposta (getRespuesta).....	7
2.1.4. Obtenir justificant (getJustificante).....	7
2.1.5. Model de classes dels serveis SCSP.....	8
2.2. Servei d'informes.....	9
2.2.1. Informe de procediments.....	9
2.2.2. Informe d'usuaris.....	10
2.2.3. Informe de serveis.....	10
2.2.4. Informe general d'estat.....	11
2.3. Servei d'estadístiques.....	12
2.3.1. Estadística de consultes.....	12
2.3.2. Estadística de càrrega.....	13
2.4. Servei de consulta de dades obertes.....	15
2.5. Documentació a la pròpia aplicació.....	16
<b>3. Client Java.....</b>	<b>17</b>
3.1. Serveis SCSP.....	18
3.1.1. Consulta de datos de identidad (SVDDGPCIWS02).....	18
3.1.2. Verificación de datos de identidad (SVDDGPVIWS02).....	20
3.1.3. Estar al corriente de obligaciones tributarias para solicitud de subvenciones y ayudas de la CCAA (SVDCCAACPASWS01).....	22
3.2. Servei d'informes.....	24
3.3. Servei d'estadístiques.....	25
3.4. Servei de dades obertes.....	26

## 1. Objecte

En aquest document es proporciona tota la informació necessària per a la integració d'una aplicació amb el l'API REST PINBAL.

## 2. Serveis disponibles

### 2.1. Serveis SCSP

La URL base per a les consultes a aquest servei és la següent:

[http://PINBAL\\_SERVER/pinbalapi/interna/recobriment](http://PINBAL_SERVER/pinbalapi/interna/recobriment)

Tots aquests serveis SCSP tenen una estructura de mètodes idèntica que detallam a continuació.

#### 2.1.1. Petició síncrona (peticionSincrona)

Aquesta petició permet l'enviament d'una petició SCSP de forma síncrona. Això vol dir que el servei esperarà fins que la petició s'hagi processat completament abans de retornar la resposta.

Aquest mètode requereix com a paràmetre un objecte de tipus **Peticion** i retorna un objecte de tipus **Respuesta**.

L'objecte de tipus Peticion ha de complir les següents restriccions:

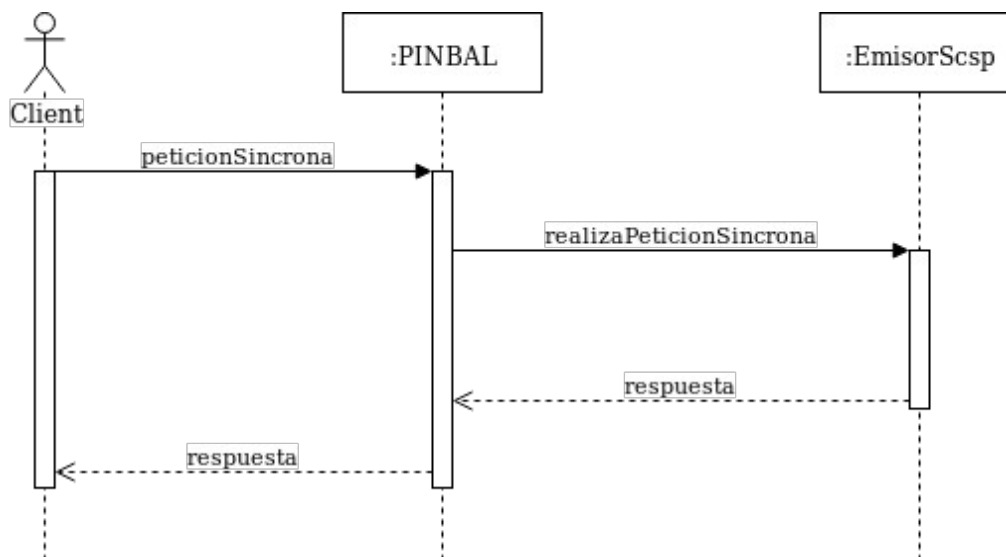
1. Ha d'estar informat l'objecte **Atributos** amb els camps *codigoCertificado* i *numElementos* emplenats.
2. En les peticions síncrones només es pot enviar una sol·licitud a dins cada petició i el valor del camp *numElementos* ha de ser 1.
3. Els altres camps de l'objecte **Atributos** s'ignoraran i es poden deixar en blanc.
4. El camp *solicitudes* ha de contenir una única **SolicitudTransmision** i aquesta s'ha d'emplenar.

Per a més detalls sobre com emplenar la informació de la sol·licitud es pot consultar l'Annex I i la documentació del servei SCSP que es vol invocar. L'estructura del camp **DatosEspecificos** i els camps obligatoris i opcionals poden variar d'un servei a un altre.

La informació resultant serà un objecte de tipus **Respuesta** que contendrà la informació retornada per l'emissor SCSP. Aquesta resposta contendrà un objecte de tipus **Atributos** amb els camps *idPeticion* i *estado* emplenats.

El camp *estado* contendrà l'estat resultant de la petició, que normalment serà Tramitada (codi «0003»).

El camp *idPetición* conté l'identificador assignat per PINBAL a la petició. Aquest camp es podrà utilitzar per a obtenir de nou la informació de la resposta (mitjançant el mètode **getRespuesta**) o per a generar el justificant de la petició (mitjançant el mètode **getJustificante**).



### 2.1.2. Petició asíncrona (peticionAsincrona)

Aquesta petició permet l'enviament d'una petició SCSP de forma asíncrona. Això vol dir que aquest mètode retornarà abans que la resposta a la petició estigui disponible. Per a obtenir la resposta s'haurà d'anar consultant l'estat de la petició mitjançant el mètode **getRespuesta** d'aquest mateix servei.

Aquest mètode requereix com a paràmetre un objecte de tipus **Peticion** i retorna un objecte de tipus **ConfirmacionPeticion**.

L'objecte de tipus **Peticion** ha de complir les següents restriccions:

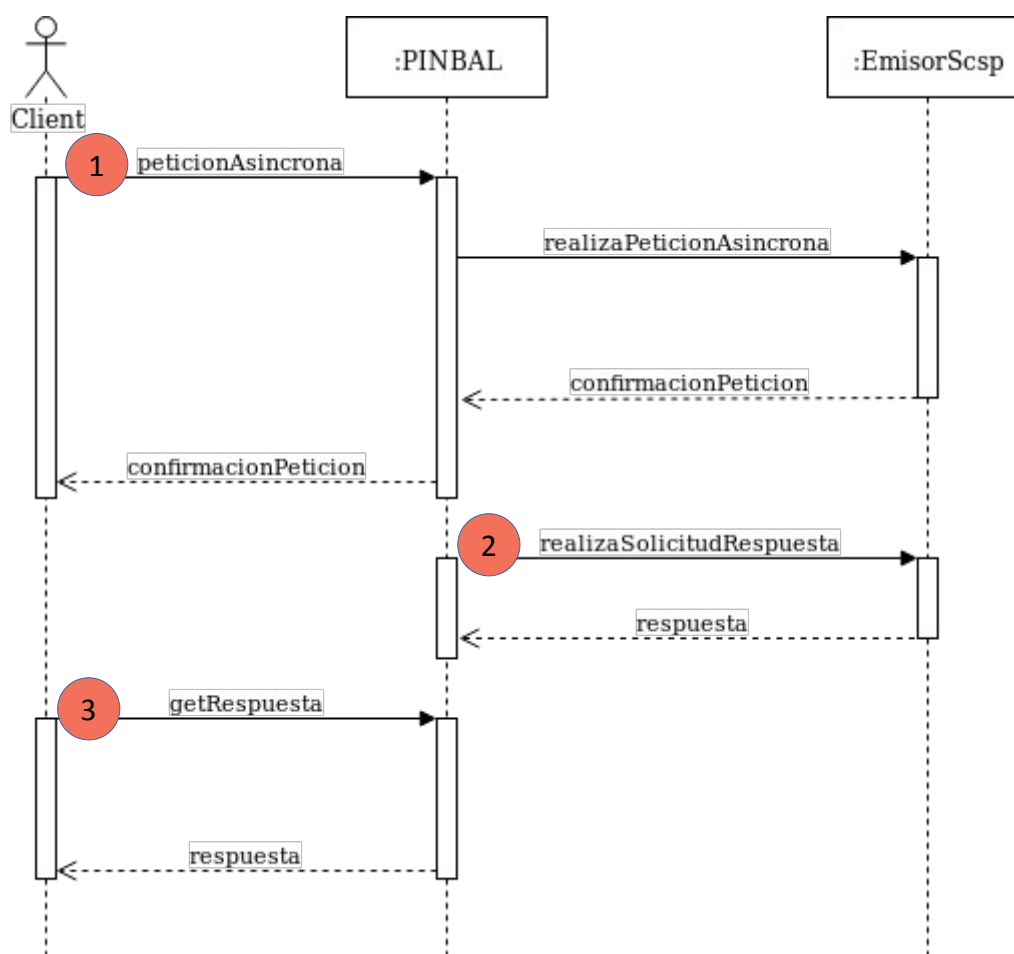
1. Ha d'estar informat l'objecte **Atributos** amb els camps *codigoCertificado* i *numElementos* emplenats.
2. En les peticions asíncrones es poden enviar múltiples sol·licituds a dins cada petició i el valor del camp *numElementos* ha de ser major que 1.
3. Els altres camps de l'objecte **Atributos** s'ignoraran i es poden deixar en blanc.
4. El camp *solicitudes* ha de contenir, com a mínim, una **SolicitudTransmission**.

La informació resultant serà un objecte de tipus **ConfirmacionPeticion**. Aquesta resposta contendrà un objecte de tipus **Atributos** amb els camps *idPetición* i *estado* emplenats.

El camp *estado* contendrà l'estat resultant de la petició, que normalment serà Tramitada (codi «0003»).

El camp *idPetición* conté l'identificador assignat per PINBAL a la petició. Aquest camp es podrà utilitzar per a obtenir de nou la informació de la resposta (mitjançant el mètode **getRespuesta**) o per a generar el justificant de la petició (mitjançant el mètode **getJustificante**).

Per a obtenir la resposta en les peticions asíncrones s'ha d'anar consultant l'estat de la petició mitjançant el mètode **getRespuesta**. Si l'estat retornat és «0003» això voldrà dir que la petició s'ha processat correctament i la mateixa resposta contendrà la informació retornada per l'emissor SCSP. Si l'estat retornat és diferent de «0003» voldrà dir que la petició encara no ha estat processada.



La seqüència d'accions per a les peticions asíncrones és la següent:

1.- El client envia la petició invocant el mètode *peticionAsincrona* del servei de recobriment. Aquest mètode, en lloc de retornar un objecte de tipus *Respuesta*, retorna un objecte de tipus *ConfirmacionPetición* amb la següent informació:

- *idPetición*: L'identificador assignat a la petició SCSP.
- *estado*: L'estat de la resposta (típicament «0002» que indica que la petició s'ha iniciat correctament i que està pendent d'obtenir la resposta).

2.- De forma periòdica, PINBAL anirà fent peticions a l'emissor SCSP. Si la resposta ja s'ha acabat de processar, l'emissor retornarà un missatge amb l'estat «0003» i la resposta SCSP. Si la resposta encara està pendent de processar, l'emissor retornarà un missatge SCSP amb l'estat «0002». PINBAL continuarà fent peticions a l'emissor SCSP fins a obtenir la resposta.

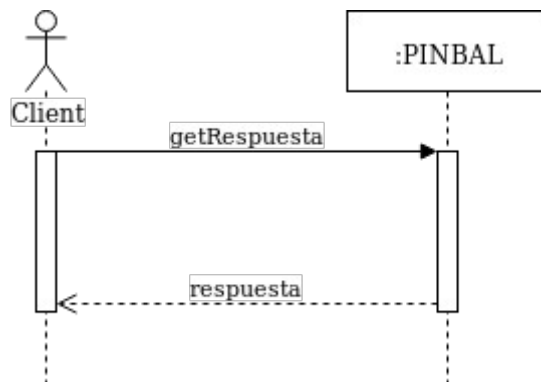
3.- El client del servei de recobriment pot anar fent peticions a PINBAL per a obtenir informació sobre l'estat de la resposta mitjançant el mètode `getRespuesta`. Si la resposta està disponible es retornarà un missatge amb l'estat «0003» que contendrà la informació retornada pel servei SCSP. Si la resposta encara no està disponible es retornarà un missatge amb l'estat «0002».

### 2.1.3. Obtenir resposta (`getRespuesta`)

Aquest mètode permet consultar l'estat d'una petició i obtenir la resposta si aquesta es troba disponible.

Aquest mètode requereix com a paràmetre un *String* amb l'identificador de la petició que es vol consultar (camp *idPetición* de l'objecte *Atributos*) i retorna un objecte de tipus **Respuesta**.

Aquest objecte de tipus **Respuesta** es retornarà sempre amb el camp *Atributos* emplenat. El camp *Transmisiones* només s'emplenarà si la petició està en estat *Tramitada* (per a més informació sobre els estats es pot consultar l'Annex I).



### 2.1.4. Obtenir justificant (`getJustificante`)

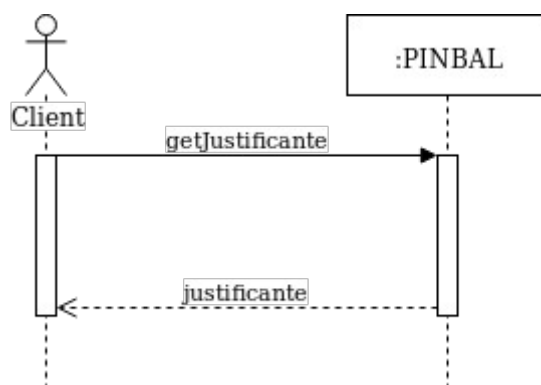
Aquest mètode retorna el contingut del justificant d'una sol·licitud en forma de document PDF firmat.

Aquest mètode accepta dos paràmetres (*idPetición* i *idSolicitud*) i retorna un array de bytes que conté un document PDF amb el justificant.

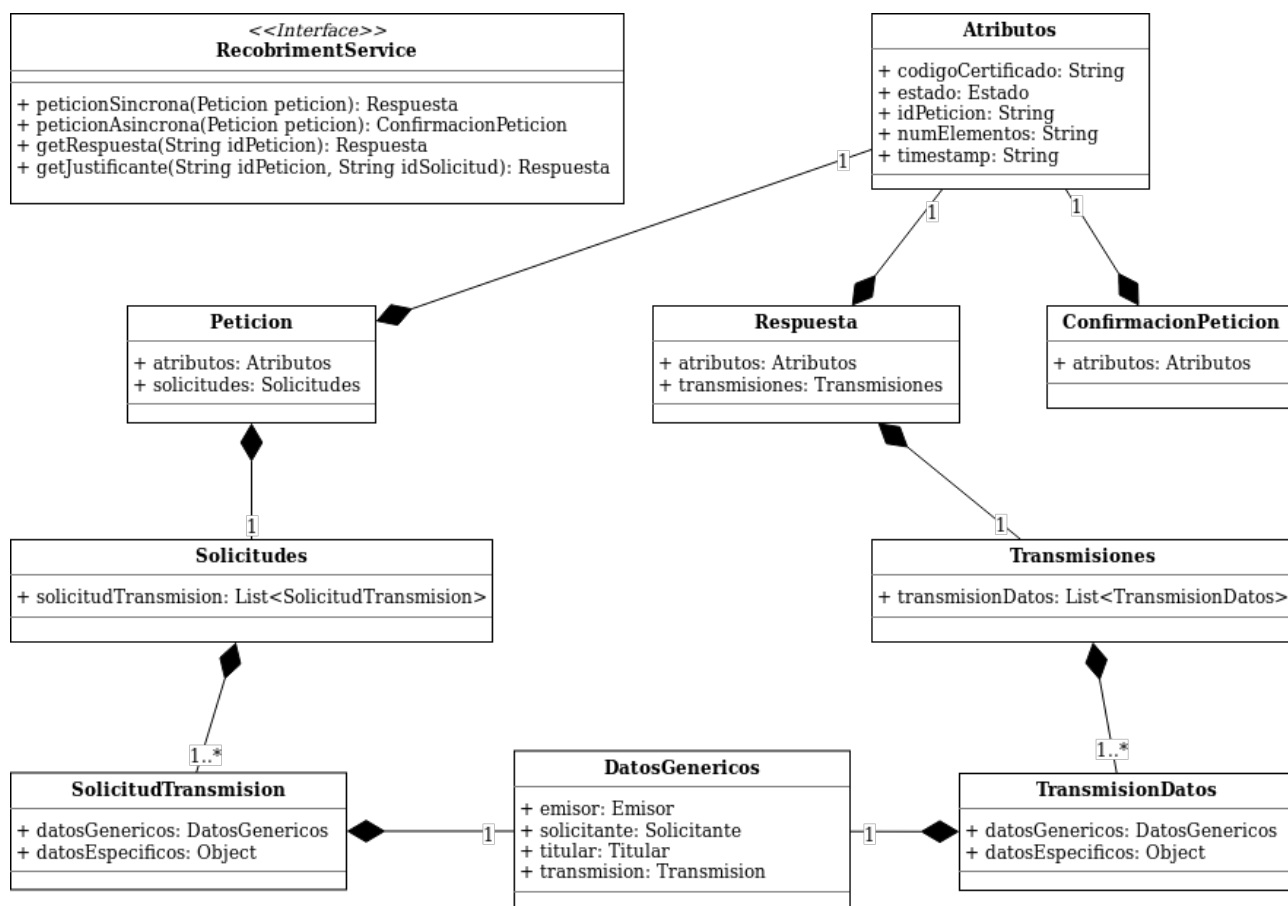
Els justificants es generen a partir d'una de les sol·licituds d'una petició i, per tant, és necessari especificar dos paràmetres: *idPetición* i *idSolicitud*. També es requereix que la petició es trobi en

estat tramitada (codi «0003»). Si la petició no es troba en aquest estat aquest mètode retornarà un error.

En el cas particular de que la petició només contengui una única sol·licitud, només serà necessari emplenar el paràmetre *idPetición* per a generar el justificant (el paràmetre *idSolicitud* es pot deixar amb el valor null).



## 2.1.5. Model de classes dels serveis SCSP





## 2.2. Servei d'informes

Aquest servei permet consultar la mateixa informació que hi ha disponible en la secció d'informes de la interfície web.

La URL base per a les consultes a aquest servei és la següent:

[http://PINBAL\\_SERVER/pinbalapi/externa/reports](http://PINBAL_SERVER/pinbalapi/externa/reports)

### 2.2.1. Informe de procediments

Informe de procediments	
<b>Adreça del servei</b>	pinbalapi/externa/reports/procediments
<b>Descripció</b>	Aquest servei retorna una llista dels procediments disponibles a l'aplicació agrupats per entitat i departament. També s'indica si el procediment està actiu o no.
<b>Mètode</b>	GET
<b>Rol requerit</b>	PBL_REPORT
<b>Paràmetres de la petició</b>	-
<b>Exemple de resposta</b>	<pre>{   "entitats": [     {       "codi": "ENT",       "nom": "Entitat",       "nif": "12345678Z",       "departaments": [         {           "codi": "DEPT",           "nom": "Departament",           "procediments": [             {               "codi": "PROC1",               "nom": "Procediment 1",               "actiu": true             }, {               "codi": "PROC2",               "nom": "Procediment 2",               "actiu": true             }           ]         }       ]     }   ] }</pre>

## 2.2.2. Informe d'usuaris

Informe d'usuaris	
<b>Adreça del servei</b>	pinbalapi/externa/reports/usuaris
<b>Descripció</b>	Aquest servei retorna una llista dels usuaris disponibles a l'aplicació agrupats per entitat i departament.
<b>Mètode</b>	GET
<b>Rol requerit</b>	PBL_REPORT
<b>Paràmetres de la petició</b>	-
<b>Exemple de resposta</b>	<pre> {   "entitats": [     {       "codi": "ENT",       "nom": "Entitat",       "nif": "12345678Z",       "departaments": [         {           "codi": "DEPT",           "nom": "Departament",           "usuaris": [             {               "codi": "USU1",               "nom": "Usuari",               "nif": "1",               "12345678Z"             }, {               "codi": "USU2",               "nom": "Usuari",               "nif": "2",               "12345678Z"             }           ]         }       ]     }   ] }</pre>

## 2.2.3. Informe de serveis

Informe de serveis	
<b>Adreça del servei</b>	pinbalapi/externa/reports/serveis
<b>Descripció</b>	Aquest servei retorna una llista dels serveis disponibles a l'aplicació.
<b>Mètode</b>	GET

<b>Rol requerit</b>	PBL_REPORT
<b>Paràmetres de la petició</b>	-
<b>Exemple de resposta</b>	<pre>{   "serveis": [     {       "codi": "SERV1",       "nom": "Servei 1",       "emisor": "AEAT"     }, {       "codi": "SERV2",       "nom": "Servei 2",       "emisor": "AEAT"     }   ] }</pre>

## 2.2.4. Informe general d'estat

Informe general d'estat	
<b>Adreça del servei</b>	pinbalapi/externa/reports/general
<b>Descripció</b>	Aquest servei retorna l'acumulat de peticions realitzades entre dues dates distingint entre les peticions processades correctament o amb errors.
<b>Mètode</b>	GET
<b>Rol requerit</b>	PBL_REPORT
<b>Paràmetres de la petició</b>	<ul style="list-style-type: none"> <li><b>dataInici</b>: data inicial de la consulta en format dd/MM/yyyy (obligatori)</li> <li><b>dataFi</b>: data final de la consulta en format dd/MM/yyyy (obligatori)</li> </ul>
<b>Exemple de resposta</b>	<pre>{   "entitats": [     {       "codi": "ENT",       "nom": "Entitat",       "nif": "12345678Z",       "departaments": [         {           "codi": "DEPT",           "nom": "Departament",           "procediments": [             {               "codi": "PROC",               "nom": "Procediment",               "serveis": [                 { </pre>

```

    "codi": "SERV",
    "nom": "Servei",
    "emisor": "AEAT",
    "usuarisAmbPermisos": 17,
    "consultesOk": 17,
    "consultesError": 17
  }
}

```

### 2.3. Servei d'estadístiques

Aquest servei permet consultar la mateixa informació que hi ha disponible en la secció d'estadístiques de la interfície web i també una estadística addicional que informa de la càrrega del sistema.

La URL base per a les consultes a aquest servei és la següent:

http://PINBAL\_SERVER/pinbalapi/externa/stats

### 2.3.1. Estadística de consultas

Estadístiques de consultes	
<b>Adreça del servei</b>	pinbalapi/externa/stats/consultes
<b>Descripció</b>	Aquest servei retorna una l'acumulat de peticions agrupades per procediment i servei distingint entre les peticions realitzades via interfície web o via recobriment i també si hi ha hagut errors o no.
<b>Mètode</b>	GET
<b>Rol requerit</b>	PBL_REPORT
<b>Paràmetres de la petició</b>	<ul style="list-style-type: none"> <li>entitatCodi: Codi de l'entitat (obligatori)</li> <li>prodecimentCodi: Codi del procediment (opcional)</li> <li>serveiCodi: Codi del servei (opcional)</li> <li>estatCodi: Codi de l'estat (opcional)</li> </ul>

	<ul style="list-style-type: none"> <li>agrupacioTipus: Tipus d'agrupació (proc-servei o servei-proc) (opcional)</li> <li>dataInici: data inicial de la consulta en format dd/MM/yyyy (opcional)</li> <li>dataFi: data final de la consulta en format dd/MM/yyyy (opcional)</li> </ul>
Exemple de resposta	<pre>{   "procediments": [     {       "codi": "PROC",       "nom": "Procediment",       "serveis": [         {           "codi": "SERV",           "nom": "Servei",           "emisor": "AEAT",           "consultesweb": {             "ok": 17,             "error": 17           },           "consultesRecobriment": {             "ok": 17,             "error": 17           },           "consultesTotal": 17         }       ]     },     {       "consultesweb": {         "ok": 17,         "error": 17       },       "consultesRecobriment": {         "ok": 17,         "error": 17       },       "consultesTotal": 17     }   ] }</pre>

### 2.3.2. Estadística de càrrega

Estadístiques de càrrega	
Adreça del servei	pinbalapi/externa/stats/carrega
Descripció	Aquest servei retorna una l'acumulat de peticions agrupades per procediment i servei distingint entre les peticions realitzades via interfície web o via recobriment i també si hi ha hagut errors o no.
Mètode	GET
Rol requerit	PBL_REPORT
Paràmetres de la petició	-

## Exemple de resposta

```
{
  "entitats": [
    {
      "codi": "ENT",
      "nom": "Entitat",
      "nif": "12345678Z",
      "departaments": [
        {
          "codi": "DEPT",
          "nom": "Departament",
          "procediments": [
            {
              "codi": "PROC",
              "nom": "Procediment",
              "serveis": [
                {
                  "codi": "SERV",
                  "nom": "Servei",
                  "emisor": "AEAT",
                  "totalWeb": {
                    "any": 12,
                    "mes": 12,
                    "dia": 12,
                    "hora": 12,
                    "minut": 12
                  }
                }
              ]
            }
          ]
        }
      ]
    }
  ],
  "totalRecobriment": {
    "any": 12,
    "mes": 12,
    "dia": 12,
    "hora": 12,
    "minut": 12
  },
  "totalRecobriment": {
    "any": 12,
    "mes": 12,
    "dia": 12,
    "hora": 12,
    "minut": 12
  }
}
```

	<pre> 12, 12 "hora": "minut":      }      }      },     "totalWeb": {       "any": 12,       "mes": 12,       "dia": 12,       "hora": 12,       "minut": 12     },     "totalRecobriment": {       "any": 12,       "mes": 12,       "dia": 12,       "hora": 12,       "minut": 12     }   },   "totalWeb": {     "any": 12,     "mes": 12,     "dia": 12,     "hora": 12,     "minut": 12   },   "totalRecobriment": {     "any": 12,     "mes": 12,     "dia": 12,     "hora": 12,     "minut": 12   } }, "totalWeb": {   "any": 12,   "mes": 12,   "dia": 12,   "hora": 12,   "minut": 12 }, "totalRecobriment": {   "any": 12,   "mes": 12,   "dia": 12,   "hora": 12,   "minut": 12 } } </pre>
--	---

## 2.4. Servei de consulta de dades obertes

Dades obertes	
<b>Adreça del servei</b>	pinbalapi/externa/opendata
<b>Descripció</b>	Aquest servei retorna un llistat de peticions realitzades des de

	PINBAL segons el format requerit pel portal Open Data de la CAIB.
<b>Mètode</b>	GET
<b>Rol requerit</b>	- (sense autenticació)
<b>Paràmetres de la petició</b>	<ul style="list-style-type: none"> <li>entitatCodi: Codi de l'entitat (opcional)</li> <li>dataInici: data inicial de la consulta en format dd/MM/yyyy (obligatori)</li> <li>dataFi: data final de la consulta en format dd/MM/yyyy (obligatori)</li> <li>procedimentCodi: Codi del procediment (opcional)</li> <li>serveiCodi: Codi del servei (opcional)</li> </ul>
<b>Exemple de resposta</b>	<pre>[{   entitatCodi: 'ENT',   entitatNom: 'Entitat',   entitatNif: '12345678Z',   departamentCodi: 'DEPT',   departamentNom: 'Departament',   procedimentCodi: 'PROC',   procedimentNom: 'Procediment',   emisor: 'AEAT',   serveiCodi: 'SERV1',   serveiNom: 'Servei',   data: 'dd/MM/yyyy HH:mm:ss',   tipus: 'web/recobriment',   resultat: 'ok/proces/error' }]</pre>

## 2.5. Documentació a la pròpia aplicació

La documentació de totes les operacions disponibles via API REST es pot consultar a la pròpia aplicació accedint a les següents adreces:

[https://PINBAL\\_HOST:PORT/pinbalapi/externa/api/rest](https://PINBAL_HOST:PORT/pinbalapi/externa/api/rest)

[https://PINBAL\\_HOST:PORT/pinbalapi/interna/api/rest](https://PINBAL_HOST:PORT/pinbalapi/interna/api/rest)

Allà on PINBAL\_HOST és l'adreça de l'aplicació, i PORT el port d'aquesta.

Ex. <https://proves.caib.es/pinbalapi/externa/api/rest>

<https://proves.caib.es/pinbalapi/interna/api/rest>



## API Consulta PINBAL

API EXTERNA de Consulta de PINBAL

[Contactar amb el desenvolupador](#)

### opendata

Mostrar/Amagar | Llista Operacions | Expandir Operacions

GET /opendata Informe de procediments agrupats per entitat i departament

### stats

Mostrar/Amagar | Llista Operacions | Expandir Operacions

GET /stats/carrega carrega

GET /stats/consultes consultes

### reports

Mostrar/Amagar | Llista Operacions | Expandir Operacions

GET /reports/general general

GET /reports/procediments procediments

GET /reports/serveis serveis

GET /reports/usuarios usuarios

[ BASE URL: /pinbalapi/externa ]

## API Consulta PINBAL

API INTERNA de PINBAL

[Contactar amb el desenvolupador](#)

### recobrimient

Mostrar/Amagar | Llista Operacions | Expandir Operacions

GET /recobrimient/getJustificante Justificant de petició de tipus SCSP

GET /recobrimient/getRespuesta Informe de petició de tipus SCSP

POST /recobrimient/peticionAsincrona Informe de petició asíncrona de tipus SCSP

POST /recobrimient/peticionSincrona Informe de petició síncrona de tipus SCSP

[ BASE URL: /pinbalapi/interna ]

### 3. Client Java

Amb l'objectiu de facilitar l'accés als serveis publicats a l'API REST de PINBAL s'ha desenvolupat un client en llenguatge de programació Java. Per a facilitar l'accés a les classes del client s'ha publicat al repositori Maven del Govern de les Illes Balears.

Per a utilitzar aquest client des d'un projecte Maven es pot emprar la següent dependència:

```

<dependency>
  <groupId>es.caib.pinbal</groupId>
  <artifactId>pinbal-client</artifactId>
  <version>1.4.32</version>
</dependency>
  
```

Aquesta dependència està accessible des dels repositori Maven GovernIB. Per a poder utilitzar-la serà necessari tenir definit al fitxer pom.xml del vostre projecte el següent bocí de codi XML:

```

<repositories>
  <repository>
    <id>github-governib-maven-repos</id>
    <name>GovernIB Maven Repository</name>
    <url>https://governib.github.io/maven/maven/</url>
  </repository>
</repositories>
  
```

També podeu descarregar directament la llibreria del client des de la següent adreça:

<https://github.com/GovernIB/maven/raw/gh-pages/maven/es/caib/pinbal/pinbal-client/1.4.32/pinbal-client-1.4.32.jar>

#### 3.1. Serveis SCSP

Els serveis SCSP disponibles en aquest moment mitjançant el client de l'API REST son els següents:

1. Consulta de datos de identidad (SVDDGPCIWS02).
2. Verificación de datos de identidad (SVDDGPVIWS02).
3. Estar al corriente de obligaciones tributarias para solicitud de subvenciones y ayudas de la CCAA (SVDCCAACPASWS01).
4. Estar al corriente de obligaciones tributarias para contratación con la CCAA (SVDCCAACPCWS01)
5. Estar al corriente de pago con la seguridad social (Q2827003ATGSS001)
6. Servei de consulta de DADES HISTÒRIQUES sobre els PADRONS MUNICIPALS (SCDHPAJU)

7. Consulta de datos de residencia legal ciudadanos d'origen extranjero - MINHAP (SVDRESLEGEXWS01)
8. Consulta de defunción - MINISTERIO DE JUSTICIA (SVDSCCDWS01)
9. Consulta de nacimiento (SVDSCCNWS01)
10. Consulta de los datos de discapacidad (SVDSCDDWS01)
11. Consulta de los datos de familia numerosa (SVDSCTFNWS01)

L'estructura de cada un d'aquests serveis és la mateixa i es detalla al capítol 2.1 d'aquest document.

A continuació detallarem la informació de la classe del client per a cada un d'aquests serveis SCSP.

### 3.1.1. Consulta de datos de identidad (SVDDGPCIWS02)

Per a fer peticions a aquest servei s'ha de crear una instància de la següent classe:

```
es.caib.pinbal.client.recobrimient.svddgpciws02.ClientSvddgpciws02
```

Per a crear una instància d'aquest client s'ha d'utilitzar el següent codi:

```
ClientSvddgpciws02 client = new ClientSvddgpciws02(  
    URL_BASE,  
    USUARI,  
    CONTRASENYA);
```

Els paràmetres necessaris per a crear la instància del client son:

- URL\_BASE: l'adreça base de l'aplicació PINBAL (per exemple: <https://proves.caib.es/pinbalapi>).
- USUARI: el codi d'un usuari amb el rol PBL\_WS.
- CONTRASENYA: la contrasenya de l'usuari.

Els mètodes disponibles en aquest client son els següents (per a més detall podeu consultar el capítol 2.1 d'aquest document):

```
public ScspRespuesta peticionSincrona(  
    List<SolicitudSvddgpciws02> solicitudes);  
  
public ScspConfirmacionPetition peticionAsincrona(  
    List<SolicitudSvddgpciws02> solicitudes);  
  
public ScspRespuesta getRespuesta(String idPetition);  
  
public ScspJustificante getJustificante(  
    String idPetition,  
    String idSolicitud);
```

Aquesta és l'estructura de l'objecte utilitzat per a crear les peticions a aquest servei:

SolicitudSvddgpciws02
+ identificadorSolicitante: String
+ nombreSolicitante: String
+ codigoProcedimiento: String
+ unidadTramitadora: String
+ codigoUnidadTramitadora: String
+ finalidad: String
+ consentimiento: ScspConsentimiento
+ idExpediente: String
+ funcionario: ScspFuncionario
+ titular: ScspTitular
+ numeroSoporte: String

A continuació teniu un una classe d'exemple que mostra com fer una petició a aquest servei:

```

public class Svddgpciws02Test {

    private static final String URL_BASE = "https://proves.caib.es/pinbalapi";
    private static final String USUARI = "USER";
    private static final String CONTRASENYA = "PASSWD";

    private ClientSvddgpciws02 client = new ClientSvddgpciws02(
        URL_BASE, USUARI, CONTRASENYA);

    @Test
    public void peticionSincrona() {
        SolicitudSvddgpciws02 solicitud = new SolicitudSvddgpciws02();
        solicitud.setIdentificadorSolicitante("B07167448");
        solicitud.setCodigoProcedimiento("ProvaConcepte");
        solicitud.setUnidadTramitadora("Departament de test");
        solicitud.setFinalidad("Test peticionSincrona");
        solicitud.setConsentimiento(ScspConsentimiento.Si);
        ScspFuncionario funcionario = new ScspFuncionario();
        funcionario.setNifFuncionario("00000000T");
        funcionario.setNombreCompletoFuncionario("Funcionari CAIB");
        solicitud.setFuncionario(funcionario);
        ScspTitular titular = new ScspTitular();
        titular.setTipoDocumentacion(ScspTipoDocumentacion.NIF);
        titular.setDocumentacion("12345678Z");
        solicitud.setTitular(titular);
        ScspRespuesta respuesta =
            client.peticionSincrona(Arrays.asList(solicitud));
        assertNotNull(respuesta);
    }
}

```

### 3.1.2. Verificación de datos de identidad (SVDDGPVIWS02)

Per a fer peticions a aquest servei s'ha de crear una instància de la següent classe:

```
es.caib.pinbal.client.recobrimient.svddgpviws02.ClientSvddgpviws02
```

Per a crear una instància d'aquest client s'ha d'utilitzar el següent codi:

```
ClientSvddgpviws02 client = new ClientSvddgpviws02(  
    URL_BASE,  
    USUARI,  
    CONTRASENYA);
```

Els paràmetres necessaris per a crear la instància del client son:

- URL\_BASE: l'adreça base de l'aplicació PINBAL (per exemple: <https://proves.caib.es/pinbalapi>).
- USUARI: el codi d'un usuari amb el rol PBL\_WS.
- CONTRASENYA: la contrasenya de l'usuari.

Els mètodes disponibles en aquest client son els següents (per a més detall podeu consultar el capítol 2.1 d'aquest document):

```
public ScspRespuesta peticionSincrona(  
    List<SolicitudSvddgpviws02> solicitudes);  
  
public ScspConfirmacionPeticion peticionAsincrona(  
    List<SolicitudSvddgpviws02> solicitudes);  
  
public ScspRespuesta getRespuesta(String idPeticion);  
  
public ScspJustificante getJustificante(  
    String idPeticion,  
    String idSolicitud);
```

Aquesta és l'estructura de l'objecte utilitzat per a crear les peticions a aquest servei:

SolicitudSvddgpviws02
+ identificadorSolicitante: String
+ nombreSolicitante: String
+ codigoProcedimiento: String
+ unidadTramitadora: String
+ codigoUnidadTramitadora: String
+ finalidad: String
+ consentimiento: ScspConsentimiento
+ idExpediente: String
+ funcionario: ScspFuncionario
+ titular: ScspTitular
+ fecha: Date
+ numeroSoporte: String
+ String provincia: String
+ String pais: String

A continuació teniu un una classe d'exemple que mostra com fer una petició a aquest servei:

```

public class Svddgpviws02Test {

    private static final String URL_BASE = "https://proves.caib.es/pinbalapi";
    private static final String USUARI = "USER";
    private static final String CONTRASENYA = "PASSWD";

    private ClientSvddgpviws02 client = new ClientSvddgpviws02(
        URL_BASE, USUARI, CONTRASENYA);

    @Test
    public void peticionSincrona() {
        SolicitudSvddgpviws02 solicitud = new SolicitudSvddgpviws02();
        solicitud.setIdentificadorSolicitante("B07167448");
        solicitud.setCodigoProcedimiento("ProvaConcepte");
        solicitud.setUnidadTramitadora("Departament de test");
        solicitud.setFinalidad("Test peticionSincrona");
        solicitud.setConsentimiento(ScspConsentimiento.Si);
        ScspFuncionario funcionario = new ScspFuncionario();
        funcionario.setNifFuncionario("00000000T");
        funcionario.setNombreCompletoFuncionario("Funcionari CAIB");
        solicitud.setFuncionario(funcionario);
        ScspTitular titular = new ScspTitular();
        titular.setTipoDocumentacion(ScspTipoDocumentacion.NIF);
        titular.setDocumentacion("12345678Z");
        solicitud.setTitular(titular);
        ScspRespuesta respuesta =
            client.peticionSincrona(Arrays.asList(solicitud));
        assertNotNull(respuesta);
    }
}
  
```

### 3.1.3. Estar al corriente de obligaciones tributarias para solicitud de subvenciones y ayudas de la CCAA (SVDCCAACPASWS01).

Per a fer peticions a aquest servei s'ha de crear una instància de la següent classe:

```
es.caib.pinbal.client.recobriment.svdccaacpasws01.ClientSvdccaacpasws01
```

Per a crear una instància d'aquest client s'ha d'utilitzar el següent codi:

```

ClientSvdccaacpasws01 client = new ClientSvdccaacpasws01(
    URL_BASE,
    USUARI,
    CONTRASENYA);
  
```

Els paràmetres necessaris per a crear la instància del client son:

- URL\_BASE: l'adreça base de l'aplicació PINBAL (per exemple: <https://proves.caib.es/pinbalapi>).
- USUARI: el codi d'un usuari amb el rol PBL\_WS.
- CONTRASENYA: la contrasenya de l'usuari.

Els mètodes disponibles en aquest client son els següents (per a més detall podeu consultar el capítol 2.1 d'aquest document):

```

public ScspRespuesta peticionSincrona(
    List<SolicitudSvdccaacpasws01> solicitudes);

public ScspConfirmacionPetition peticionAsincrona(
    List<SolicitudSvdccaacpasws01> solicitudes);

public ScspRespuesta getRespuesta(String idPetition);

public ScspJustificante getJustificante(
    String idPetition,
    String idSolicitud);
  
```

Aquesta és l'estructura de l'objecte utilitzat per a crear les peticions a aquest servei:

SolicitudSvdccaacpasws01
+ identificadorSolicitante: String
+ nombreSolicitante: String
+ codigoProcedimiento: String
+ unidadTramitadora: String
+ codigoUnidadTramitadora: String
+ finalidad: String
+ consentimiento: ScspConsentimiento
+ idExpediente: String
+ funcionario: ScspFuncionario
+ titular: ScspTitular
+ codigoProvincia: String
+ codigoComunidadAutonoma: String

A continuació teniu un una classe d'exemple que mostra com fer una petició a aquest servei:

```

public class Svdccaacpasws01Test {

    private static final String URL_BASE = "http://localhost:8180/pinbalapi";
    private static final String USUARI = "USER";
    private static final String CONTRASENYA = "PASSWD";

    private final ClientSvdccaacpasws01 client = new ClientSvdccaacpasws01(
        URL_BASE, USUARI, CONTRASENYA);

    @Test
    public void peticionSincrona() throws UniformInterfaceException,
        ClientHandlerException, IOException {
        SolicitudSvdccaacpasws01 solicitud = new SolicitudSvdccaacpasws01();
        solicitud.setIdentificadorSolicitante("B07167448");
        solicitud.setCodigoProcedimiento("CODSVDR_GBA_20121107");
        solicitud.setUnidadTramitadora("Departament de test");
        solicitud.setFinalidad("Test peticionSincrona");
        solicitud.setConsentimiento(ScspConsentimiento.Si);
        ScspFuncionario funcionario = new ScspFuncionario();
  
```

```

funcionario.setNifFuncionario("00000000T");
funcionario.setNombreCompletoFuncionario("Funcionari CAIB");
solicitud.setFuncionario(funcionario);
ScspTitular titular = new ScspTitular();
titular.setTipoDocumentacion(ScspTipoDocumentacion.NIF);
titular.setDocumentacion("12345678Z");
solicitud.setTitular(titular);
ScspRespuesta respuesta =
    client.peticionSincrona(Arrays.asList(solicitud));
assertNotNull(respuesta);
}
}

```

### 3.1.4. Estar al corriente de obligaciones tributarias para contratación con la CCAA (SVDCCAACPCWS01).

Per a fer peticions a aquest servei s'ha de crear una instància de la següent classe:

```
es.caib.pinbal.client.recobrimient.svdccaacpcws01.ClientSvdccaacpcws01
```

Per a crear una instància d'aquest client s'ha d'utilitzar el següent codi:

```

ClientSvdccaacpcws01 client = new ClientSvdccaacpcws01(
    URL_BASE,
    USUARI,
    CONTRASENYA);

```

Els paràmetres necessaris per a crear la instància del client son:

- URL\_BASE: l'adreça base de l'aplicació PINBAL (per exemple: <https://proves.caib.es/pinbalapi>).
- USUARI: el codi d'un usuari amb el rol PBL\_WS.
- CONTRASENYA: la contrasenya de l'usuari.

Els mètodes disponibles en aquest client son els següents (per a més detall podeu consultar el capítol 2.1 d'aquest document):

```

public ScspRespuesta peticionSincrona(
    List<SolicitudSvdccaacpcws01> solicitudes);

public ScspConfirmacionPeticion peticionAsincrona(
    List<SolicitudSvdccaacpcws01> solicitudes);

public ScspRespuesta getRespuesta(String idPeticion);

public ScspJustificante getJustificante(
    String idPeticion,
    String idSolicitud);

```



Aquesta és l'estructura de l'objecte utilitzat per a crear les peticions a aquest servei:

Classname
+ identificadorSolicitante: String
+ nombreSolicitante: String
+ codigoProcedimiento: String
+ unidadTramitadora: String
+ codigoUnidadTramitadora: String
+ finalidad: String
+ consentimiento: ScspConsentimiento
+ idExpediente: String
+ funcionario: ScscFuncionario
+ titular: ScspTitular

A continuació teniu un una classe d'exemple que mostra com fer una petició a aquest servei:

```
public class Svdccaacpcws01Test {

    private static final String URL_BASE = "https://proves.caib.es/pinbalapi";
    private static final String USUARI = "xxxxx";
    private static final String CONTRASENYA = "xxxxx";

    private ClientQSvdccaacpcws01 client = new ClientSvdccaacpcws01(
        URL_BASE, USUARI, CONTRASENYA);

    @Test
    public void peticionSincrona() throws UniformInterfaceException,
        ClientHandlerException, IOException {
        SolicitudSvdccaacpcws01 solicitud = new SolicitudSvdccaacpcws01();
        solicitud.setIdentificadorSolicitante(ENTITAT_CIF);
        solicitud.setCodigoProcedimiento(CODIGO_PROCEDIMIENTO);
        solicitud.setUnidadTramitadora("Departament de test");
        solicitud.setFinalidad("Test peticionSincrona");
        solicitud.setConsentimiento(ScspConsentimiento.Si);
        ScspFuncionario funcionario = new ScspFuncionario();
        funcionario.setNifFuncionario("00000000T");
        funcionario.setNombreCompletoFuncionario("Funcionari CAIB");
        solicitud.setFuncionario(funcionario);
        ScspTitular titular = new ScspTitular();
        titular.setTipoDocumentacion(ScspTipoDocumentacion.NIF);
        titular.setDocumentacion("12345678Z");
        solicitud.setTitular(titular);
        solicitud.setCodigoComunidadAutonoma("04");
        solicitud.setCodigoProvincia("07");
        ScspRespuesta respuesta =
            client.peticionSincrona(Arrays.asList(solicitud));
        assertNotNull(respuesta);
    }
}
```

### 3.1.5. Estar al corriente de pago con la seguridad social (Q2827003ATGSS001)

Per a fer peticions a aquest servei s'ha de crear una instància de la següent classe:

```
es.caib.pinbal.client.recobrimient.q2827003atgss001.ClientQ2827003atgss001
```

Per a crear una instància d'aquest client s'ha d'utilitzar el següent codi:

```
ClientQ2827003atgss001 client = new ClientQ2827003atgss001(  
    URL_BASE,  
    USUARI,  
    CONTRASENYA);
```

Els paràmetres necessaris per a crear la instància del client son:

- URL\_BASE: l'adreça base de l'aplicació PINBAL (per exemple: <https://proves.caib.es/pinbalapi>).
- USUARI: el codi d'un usuari amb el rol PBL\_WS.
- CONTRASENYA: la contrasenya de l'usuari.

Els mètodes disponibles en aquest client son els següents (per a més detall podeu consultar el capítol 2.1 d'aquest document):

```
public ScspRespuesta peticionSincrona(  
    List<SolicitudQ2827003atgss001> solicitudes);  
  
public ScspConfirmacionPeticon peticionAsincrona(  
    List<SolicitudQ2827003atgss001> solicitudes);  
  
public ScspRespuesta getRespuesta(String idPeticon);  
  
public ScspJustificante getJustificante(  
    String idPeticon,  
    String idSolicitud);
```

Aquesta és l'estructura de l'objecte utilitzat per a crear les peticions a aquest servei:

SolicitudQ2827003atgss001
+ identificadorSolicitante: String
+ nombreSolicitante: String
+ codigoProcedimiento: String
+ unidadTramitadora: String
+ codigoUnidadTramitadora: String
+ finalidad: String
+ consentimiento: ScspConsentimiento
+ idExpediente: String
+ funcionario: ScscFuncionario
+ titular: ScspTitular
+ codigoProvincia: String
+ codigoComunidadAutonoma: String

A continuació teniu un una classe d'exemple que mostra com fer una petició a aquest servei:

```
public class Q2827003atgss001Test {

    private static final String URL_BASE = "https://proves.caib.es/pinbalapi";
    private static final String USUARI = "xxxxx";
    private static final String CONTRASENYA = "xxxxx";

    private ClientQ2827003atgss001 client = new ClientQ2827003atgss001(
        URL_BASE, USUARI, CONTRASENYA);

    @Test
    public void peticionSincrona() throws UniformInterfaceException,
        ClientHandlerException, IOException {
        SolicitudQ2827003atgss001 solicitud = new SolicitudQ2827003atgss001();
        solicitud.setIdentificadorSolicitante(ENTITAT_CIF);
        solicitud.setCodigoProcedimiento(CODIGO_PROCEDIMIENTO);
        solicitud.setUnidadTramitadora("Departament de test");
        solicitud.setFinalidad("Test peticionSincrona");
        solicitud.setConsentimiento(ScspConsentimiento.Si);
        ScspFuncionario funcionario = new ScspFuncionario();
        funcionario.setNifFuncionario("00000000T");
        funcionario.setNombreCompletoFuncionario("Funcionari CAIB");
        solicitud.setFuncionario(funcionario);
        ScspTitular titular = new ScspTitular();
        titular.setTipoDocumentacion(ScspTipoDocumentacion.DNI);
        titular.setDocumentacion("12345678Z");
        solicitud.setTitular(titular);
        ScspRespuesta respuesta =
            client.peticionSincrona(Arrays.asList(solicitud));
        assertNotNull(respuesta);
    }
}
```

### 3.1.6. Servei de consulta de DADES HISTÒRIQUES sobre els PADRONS MUNICIPALS (SCDHPAJU).

Per a fer peticions a aquest servei s'ha de crear una instància de la següent classe:

```
es.caib.pinbal.client.recobrimient.scdhpaju.ClientScdhpaju
```

Per a crear una instància d'aquest client s'ha d'utilitzar el següent codi:

```
ClientScdhpaju client = new ClientScdhpaju(
    URL_BASE,
    USUARI,
    CONTRASENYA);
```

Els paràmetres necessaris per a crear la instància del client son:

- URL\_BASE: l'adreça base de l'aplicació PINBAL (per exemple: <https://proves.caib.es/pinbalapi>).
- USUARI: el codi d'un usuari amb el rol PBL\_WS.

- CONTRASENYA: la contrasenya de l'usuari.

Els mètodes disponibles en aquest client son els següents (per a més detall podeu consultar el capítol 2.1 d'aquest document):

```
public ScspRespuesta peticionSincrona(
    List<SolicitudScdhpaju> solicitudes);

public ScspConfirmacionPeticion peticionAsincrona(
    List<SolicitudScdhpaju> solicitudes);

public ScspRespuesta getRespuesta(String idPeticion);

public ScspJustificante getJustificante(
    String idPeticion,
    String idSolicitud);
```

Aquesta és l'estructura de l'objecte utilitzat per a crear les peticions a aquest servei:

SolicitudScdhpaju
+ identificadorSolicitante: String
+ nombreSolicitante: String
+ codigoProcedimiento: String
+ unidadTramitadora: String
+ codigoUnidadTramitadora: String
+ finalidad: String
+ consentimiento: ScspConsentimiento
+ idExpediente: String
+ funcionario: ScspFuncionario
+ titular: ScspTitular
+ provinciaSolicitud: String
+ municipioSolicitud: String
+ documentacion_tipo: String
+ documentacion_valor: String
+ documentacion_numSoporte: String
+ datosPersonales_tipo: String
+ datosPersonales_valor: String
+ datosPersonales_numSoporte: String
+ nombre: String
+ particula1: String
+ apellido1: String
+ particula2: String
+ apellido2: String
+ fechaNacimiento: String
+ nia: String
+ numeroAnyos: String

A continuació teniu un una classe d'exemple que mostra com fer una petició a aquest servei:

```
public class ScdhpajuTest {

    private static final String URL_BASE = "https://proves.caib.es/pinbalapi";
    private static final String USUARI = "xxxxxx";
    private static final String CONTRASENYA = "xxxxxx";
```

```
private ClientScdhpaju client = new ClientScdhpaju(
    URL_BASE, USUARI, CONTRASENYA);

@Test
public void peticionSincrona() throws UniformInterfaceException,
ClientHandlerException, IOException {
    SolicitudScdhpaju solicitud = new SolicitudScdhpaju();
    solicitud.setIdentificadorSolicitante(ENTITAT_CIF);
    solicitud.setCodigoProcedimiento(CODIGO_PROCEIMIENTO);
    solicitud.setUnidadTramitadora("Departament de test");
    solicitud.setFinalidad("Test peticionSincrona");
    solicitud.setConsentimiento(ScspConsentimiento.Si);
    ScspFuncionario funcionario = new ScspFuncionario();
    funcionario.setNifFuncionario("00000000T");
    funcionario.setNombreCompletoFuncionario("Funcionari CAIB");
    solicitud.setFuncionario(funcionario);
    ScspTitular titular = new ScspTitular();
    titular.setTipoDocumentacion(ScspTipoDocumentacion.DNI);
    titular.setDocumentacion("12345678Z");
    solicitud.setTitular(titular);

    solicitud.setConsultaPerDocumentIdentitat("NIF", "12345678Z", null);
    solicitud.setConsultaPerDadesPersonals(null, null, null, null, null, null,
null, null, null);
    solicitud.setConsultaPerReferenciaNia(null);
    solicitud.setProvinciaSolicitud("07");
    solicitud.setMunicipioSolicitud("001");
    solicitud.setNumeroAnyos(null);

    ScspRespuesta respuesta =
        client.peticionSincrona(Arrays.asList(solicitud));
    assertNotNull(respuesta);
}
```

### 3.1.7. Consulta de datos de residencia legal ciudadanos d'origen estranger - MINHAP (SVDRESLEGEXWS01).

Per a fer peticions a aquest servei s'ha de crear una instància de la següent classe:

```
es.caib.pinbal.client.recobriment.svdreslegexws01.ClientSvdreslegexws01
```

Per a crear una instància d'aquest client s'ha d'utilitzar el següent codi:

```
ClientSvdreslegexws01 client = new ClientSvdreslegexws01(
    URL_BASE,
    USUARI,
    CONTRASENYA);
```

Els paràmetres necessaris per a crear la instància del client son:

- URL\_BASE: l'adreça base de l'aplicació PINBAL (per exemple: <https://proves.caib.es/pinbalapi>).

- USUARI: el codi d'un usuari amb el rol PBL\_WS.
- CONTRASENYA: la contrasenya de l'usuari.

Els mètodes disponibles en aquest client son els següents (per a més detall podeu consultar el capítol 2.1 d'aquest document):

```
public ScspRespuesta peticionSincrona(
    List<SolicitudSvdreslegexws01> solicitudes);

public ScspConfirmacionPeticon peticionAsincrona(
    List<SolicitudSvdreslegexws01> solicitudes);

public ScspRespuesta getRespuesta(String idPeticon);

public ScspJustificante getJustificante(
    String idPeticon,
    String idSolicitud);
```

Aquesta és l'estructura de l'objecte utilitzat per a crear les peticions a aquest servei:

SolicitudSvdreslegexws01
+ identificadorSolicitante: String
+ nombreSolicitante: String
+ codigoProcedimiento: String
+ unidadTramitadora: String
+ codigoUnidadTramitadora: String
+ finalidad: String
+ consentimiento: ScspConsentimiento
+ idExpediente: String
+ funcionario: ScscFuncionario
+ titular: ScspTitular
+ anioNacimiento: String
+ nacionalidad: String

A continuació teniu un una classe d'exemple que mostra com fer una petició a aquest servei:

```
public class Svdreslegexws01Test {

    private static final String URL_BASE = "https://proves.caib.es/pinbalapi";
    private static final String USUARI = "xxxxxx";
    private static final String CONTRASENYA = "xxxxxx";

    private ClientSvdreslegexws01 client = new ClientSvdreslegexws01(
        URL_BASE, USUARI, CONTRASENYA);

    @Test
    public void peticionSincrona() throws UniformInterfaceException,
        ClientHandlerException, IOException {
        SolicitudSvdreslegexws01 solicitud = new SolicitudSvdreslegexws01();
        solicitud.setIdentificadorSolicitante(ENTITAT_CIF);
        solicitud.setCodigoProcedimiento(CODIGO_PROCEDIMIENTO);
        solicitud.setUnidadTramitadora("Departament de test");
        solicitud.setFinalidad("Test peticionSincrona");
    }
}
```

```

solicitud.setConsentimiento(ScspConsentimiento.Si);
ScspFuncionario funcionario = new ScspFuncionario();
funcionario.setNifFuncionario("00000000T");
funcionario.setNombreCompletoFuncionario("Funcionari CAIB");
solicitud.setFuncionario(funcionario);
ScspTitular titular = new ScspTitular();
titular.setTipoDocumentacion(ScspTipoDocumentacion.NIE);
titular.setDocumentacion("Z4308218R");
solicitud.setTitular(titular);
solicitud.setAnioNacimiento(null);
solicitud.setNacionalidad(null);

ScspRespuesta respuesta =
    client.peticionSincrona(Arrays.asList(solicitud));
assertNotNull(respuesta);
}
}

```

### 3.1.8. Consulta de defunción - MINISTERIO DE JUSTICIA (SVDSCCDWS01).

Per a fer peticions a aquest servei s'ha de crear una instància de la següent classe:

```
es.caib.pinbal.client.recobrimient.svdscddws01.ClientSvdscddws01
```

Per a crear una instància d'aquest client s'ha d'utilitzar el següent codi:

```

ClientSvdscddws01 client = new ClientSvdscddws01(
    URL_BASE,
    USUARI,
    CONTRASENYA);

```

Els paràmetres necessaris per a crear la instància del client son:

- URL\_BASE: l'adreça base de l'aplicació PINBAL (per exemple: <https://proves.caib.es/pinbalapi>).
- USUARI: el codi d'un usuari amb el rol PBL\_WS.
- CONTRASENYA: la contrasenya de l'usuari.

Els mètodes disponibles en aquest client son els següents (per a més detall podeu consultar el capítol 2.1 d'aquest document):

```

public ScspRespuesta peticionSincrona(
    List<SolicitudSvdscddws01> solicitudes);

public ScspConfirmacionPeticion peticionAsincrona(
    List<SolicitudSvdscddws01> solicitudes);

public ScspRespuesta getRespuesta(String idPeticion);

public ScspJustificante getJustificante(
    String idPeticion,
    String idSolicitud);

```



Aquesta és l'estructura de l'objecte utilitzat per a crear les peticions a aquest servei:

SolicitudSvdsccdws01
+ identificadorSolicitante: String
+ nombreSolicitante: String
+ codigoProcedimiento: String
+ unidadTramitadora: String
+ codigoUnidadTramitadora: String
+ finalidad: String
+ consentimiento: ScspConsentimiento
+ idExpediente: String
+ funcionario: ScspFuncionario
+ titular: ScspTitular
+ fechaHechoRegistral: String
+ ausenciaSegundoApellido: Boolean
+ registroCivil: String
+ tomo: String
+ pagina: String
+ poblacionHechoRegistral: String
+ fechaNacimiento: String
+ poblacionNacimiento: String
+ nombrePadre: String
+ nombreMadre: String

A continuació teniu un una classe d'exemple que mostra com fer una petició a aquest servei:

```
public class Svdsccdws01Test {

    private static final String URL_BASE = "https://proves.caib.es/pinbalapi";
    private static final String USUARI = "xxxxxx";
    private static final String CONTRASENYA = "xxxxxx";

    private ClientSvdsccdws01 client = new ClientSvdsccdws01(
        URL_BASE, USUARI, CONTRASENYA);

    @Test
    public void peticionSincrona() throws UniformInterfaceException,
        ClientHandlerException, IOException {
        SolicitudSvdsccdws01 solicitud = new SolicitudSvdsccdws01();
        solicitud.setIdentificadorSolicitante(ENTITAT_CIF);
        solicitud.setCodigoProcedimiento(CODIGO_PROCEDIMIENTO);
        solicitud.setUnidadTramitadora("Departament de test");
        solicitud.setFinalidad("Test peticionSincrona");
        solicitud.setConsentimiento(ScspConsentimiento.Si);
        ScspFuncionario funcionario = new ScspFuncionario();
        funcionario.setNifFuncionario("00000000T");
        funcionario.setNombreCompletoFuncionario("Funcionari CAIB");
        solicitud.setFuncionario(funcionario);
        ScspTitular titular = new ScspTitular();
        titular.setTipoDocumentacion(ScspTipoDocumentacion.DNI);
        titular.setTipoDocumentacion(ScspTipoDocumentacion.DNI);
        titular.setDocumentacion("12345678Z");
        titular.setNombre("Usuario");
        titular.setApellido1("Prueba");
        solicitud.setTitular(titular);
        solicitud.setDadesAdicionalsTitular("2021-11-15", true);
        solicitud.setConsultaPerDadesRegistrals(null, null, null);
        solicitud.setConsultaPerAltresDades(null, null, null, null, null);

        ScspRespuesta respuesta =
```



```

        client.peticiónSincrona(Arrays.asList(solicitud));
        assertNotNull(respuesta);
    }
}

```

### 3.1.9. Consulta de nacimiento (SVDSCCNWS01).

Per a fer peticions a aquest servei s'ha de crear una instància de la següent classe:

```
es.caib.pinbal.client.recobrimient.svdscnws01.ClientSvdscnws01
```

Per a crear una instància d'aquest client s'ha d'utilitzar el següent codi:

```

ClientSvdscnws01 client = new ClientSvdscnws01(
    URL_BASE,
    USUARI,
    CONTRASENYA);

```

Els paràmetres necessaris per a crear la instància del client son:

- URL\_BASE: l'adreça base de l'aplicació PINBAL (per exemple: <https://proves.caib.es/pinbalapi>).
- USUARI: el codi d'un usuari amb el rol PBL\_WS.
- CONTRASENYA: la contrasenya de l'usuari.

Els mètodes disponibles en aquest client son els següents (per a més detall podeu consultar el capítol 2.1 d'aquest document):

```

public ScspRespuesta peticiónSincrona(
    List<SolicitudSvdscnws01> solicitudes);

public ScspConfirmacionPetición peticiónAsincrona(
    List<SolicitudSvdscnws01> solicitudes);

public ScspRespuesta getRespuesta(String idPetición);

public ScspJustificante getJustificante(
    String idPetición,
    String idSolicitud);

```

Aquesta és l'estructura de l'objecte utilitzat per a crear les peticions a aquest servei:

SolicitudSvdscnws01
+ identificadorSolicitante: String + nombreSolicitante: String + codigoProcedimiento: String + unidadTramitadora: String + codigoUnidadTramitadora: String + finalidad: String + consentimiento: ScspConsentimiento + idExpediente: String + funcionario: ScspFuncionario + titular: ScspTitular + fechaHechoRegistral: String + ausenciaSegundoApellido: Boolean + registroCivil: String + tomo: String + pagina: String + poblacionHechoRegistral: String + fechaNacimiento: String + poblacionNacimiento: String + nombrePadre: String + nombreMadre: String

A continuació teniu un una classe d'exemple que mostra com fer una petició a aquest servei:

```

public class Svdscnws01Test {

    private static final String URL_BASE = "https://proves.caib.es/pinbalapi";
    private static final String USUARI = "xxxxxx";
    private static final String CONTRASENYA = "xxxxxx";

    private ClientSvdscnws01 client = new ClientSvdscnws01(
        URL_BASE, USUARI, CONTRASENYA);

    @Test
    public void peticionSincrona() throws UniformInterfaceException,
        ClientHandlerException, IOException {
        SolicitudSvdscnws01 solicitud = new SolicitudSvdscnws01();
        solicitud.setIdentificadorSolicitante(ENTITAT_CIF);
        solicitud.setCodigoProcedimiento(CODIGO_PROCEDIMIENTO);
        solicitud.setUnidadTramitadora("Departament de test");
        solicitud.setFinalidad("Test peticionSincrona");
        solicitud.setConsentimiento(ScspConsentimiento.Si);
        ScspFuncionario funcionario = new ScspFuncionario();
        funcionario.setNifFuncionario("00000000T");
        funcionario.setNombreCompletoFuncionario("Funcionari CAIB");
        solicitud.setFuncionario(funcionario);
        ScspTitular titular = new ScspTitular();
        titular.setTipoDocumentacion(ScspTipoDocumentacion.DNI);
        titular.setDocumentacion("12345678Z");
        titular.setNombre("Usuario");
        titular.setApellido1("Prueba");
        solicitud.setTitular(titular);
        solicitud.setDadesAdicionalsTitular("2021-11-15", true);
        solicitud.setConsultaPerDadesRegistrals(null, null, null);
        solicitud.setConsultaPerAltresDades(null, null, null, null, null);
        ScspRespuesta respuesta =
            client.peticionSincrona(Arrays.asList(solicitud));
        assertNotNull(respuesta);
    }
}
  
```

### 3.1.10.Consulta de los datos de discapacidad (SVDSCDDWS01).

Per a fer peticions a aquest servei s'ha de crear una instància de la següent classe:

```
es.caib.pinbal.client.recobrimient.svdsdcdws01.ClientSvdsdcdws01
```

Per a crear una instància d'aquest client s'ha d'utilitzar el següent codi:

```
ClientSvdsdcdws01 client = new ClientSvdsdcdws01(  
    URL_BASE,  
    USUARI,  
    CONTRASENYA);
```

Els paràmetres necessaris per a crear la instància del client son:

- URL\_BASE: l'adreça base de l'aplicació PINBAL (per exemple: <https://proves.caib.es/pinbalapi>).
- USUARI: el codi d'un usuari amb el rol PBL\_WS.
- CONTRASENYA: la contrasenya de l'usuari.

Els mètodes disponibles en aquest client son els següents (per a més detall podeu consultar el capítol 2.1 d'aquest document):

```
public ScspRespuesta peticionSincrona(  
    List<SolicitudSvdsdcdws01> solicitudes);  
  
public ScspConfirmacionPeticion peticionAsincrona(  
    List<SolicitudSvdsdcdws01> solicitudes);  
  
public ScspRespuesta getRespuesta(String idPeticion);  
  
public ScspJustificante getJustificante(  
    String idPeticion,  
    String idSolicitud);
```

Aquesta és l'estructura de l'objecte utilitzat per a crear les peticions a aquest servei:

SolicitudSvdsdcdws01
+ identificadorSolicitante: String
+ nombreSolicitante: String
+ codigoProcedimiento: String
+ unidadTramitadora: String
+ codigoUnidadTramitadora: String
+ finalidad: String
+ consentimiento: ScspConsentimiento
+ idExpediente: String
+ funcionario: ScscFuncionario
+ titular: ScspTitular
+ codigoComunidadAutonoma: String
+ codigoProvincia: String
+ expediente: String

A continuació teniu un una classe d'exemple que mostra com fer una petició a aquest servei:

```
public class Svdsdcdws01Test {

    private static final String URL_BASE = "https://proves.caib.es/pinbalapi";
    private static final String USUARI = "xxxxx";
    private static final String CONTRASENYA = "xxxxx";

    private ClientSvdsdcdws01 client = new ClientSvdsdcdws01(
        URL_BASE, USUARI, CONTRASENYA);

    @Test
    public void peticionSincrona() throws UniformInterfaceException,
        ClientHandlerException, IOException {
        SolicitudSvdsdcdws01 solicitud = new SolicitudSvdsdcdws01();
        solicitud.setIdentificadorSolicitante(ENTITAT_CIF);
        solicitud.setCodigoProcedimiento(CODIGO_PROCEDIMIENTO);
        solicitud.setUnidadTramitadora("Departament de test");
        solicitud.setFinalidad("Test peticionSincrona");
        solicitud.setConsentimiento(ScspConsentimiento.Si);
        ScspFuncionario funcionario = new ScspFuncionario();
        funcionario.setNifFuncionario("00000000T");
        funcionario.setNombreCompletoFuncionario("Funcionari CAIB");
        solicitud.setFuncionario(funcionario);
        ScspTitular titular = new ScspTitular();
        titular.setTipoDocumentacion(ScspTipoDocumentacion.DNI);
        titular.setDocumentacion("12345678Z");
        solicitud.setTitular(titular);
        solicitud.setCodigoComunidadAutonoma("04");
        solicitud.setCodigoProvincia("07");
        solicitud.setFechaConsulta("30/12/2021");
        solicitud.setFechaNacimiento("17/08/1970");
        solicitud.setConsentimientoTiposDiscapacidad("S");
        ScspRespuesta respuesta =
            client.peticionSincrona(Arrays.asList(solicitud));
        assertNotNull(respuesta);
    }
}
```

### 3.1.11.Consulta de los datos de familia numerosa (SVDSCTFNWS01).

Per a fer peticions a aquest servei s'ha de crear una instància de la següent classe:

```
es.caib.pinbal.client.recobrimient.svdsctfnws01.ClientSvdsctfnws01
```

Per a crear una instància d'aquest client s'ha d'utilitzar el següent codi:

```
ClientSvdsctfnws01 client = new ClientSvdsctfnws01(
    URL_BASE,
    USUARI,
    CONTRASENYA);
```

Els paràmetres necessaris per a crear la instància del client son:

- URL\_BASE: l'adreça base de l'aplicació PINBAL (per exemple: <https://proves.caib.es/pinbalapi>).

- USUARI: el codi d'un usuari amb el rol PBL\_WS.
- CONTRASENYA: la contrasenya de l'usuari.

Els mètodes disponibles en aquest client son els següents (per a més detall podeu consultar el capítol 2.1 d'aquest document):

```
public ScspRespuesta peticionSincrona(
    List<SolicitudSvdsctfnws01> solicitudes);

public ScspConfirmacionPeticon peticionAsincrona(
    List<SolicitudSvdsctfnws01> solicitudes);

public ScspRespuesta getRespuesta(String idPeticon);

public ScspJustificante getJustificante(
    String idPeticon,
    String idSolicitud);
```

Aquesta és l'estructura de l'objecte utilitzat per a crear les peticions a aquest servei:

SolicitudSvdsctfnws01
+ identificadorSolicitante: String
+ nombreSolicitante: String
+ codigoProcedimiento: String
+ unidadTramitadora: String
+ codigoUnidadTramitadora: String
+ finalidad: String
+ consentimiento: ScspConsentimiento
+ idExpediente: String
+ funcionario: ScscFuncionario
+ titular: ScspTitular
+ codigoComunidadAutonoma: String
+ numeroTitulo: String

A continuació teniu un una classe d'exemple que mostra com fer una petició a aquest servei:

```
public class Svdsctfnws01Test {

    private static final String URL_BASE = "https://proves.caib.es/pinbalapi";
    private static final String USUARI = "xxxxx";
    private static final String CONTRASENYA = "xxxxx";

    private ClientSvdsctfnws01 client = new ClientSvdsctfnws01(
        URL_BASE, USUARI, CONTRASENYA);

    @Test
    public void peticionSincrona() throws UniformInterfaceException,
        ClientHandlerException, IOException {
        SolicitudSvdsctfnws01 solicitud = new SolicitudSvdsctfnws01();
        solicitud.setIdentificadorSolicitante(ENTITAT_CIF);
        solicitud.setCodigoProcedimiento(CODIGO_PROCEDIMIENTO);
        solicitud.setUnidadTramitadora("Departament de test");
        solicitud.setFinalidad("Test peticionSincrona");
    }
}
```

```
solicitud.setConsentimiento(ScspConsentimiento.Si);
ScspFuncionario funcionario = new ScspFuncionario();
funcionario.setNifFuncionario("00000000T");
funcionario.setNombreCompletoFuncionario("Funcionari CAIB");
solicitud.setFuncionario(funcionario);
ScspTitular titular = new ScspTitular();
titular.setTipoDocumentacion(ScspTipoDocumentacion.DNI);
titular.setDocumentacion("12345678Z");
solicitud.setTitular(titular);
solicitud.setCodigoComunidadAutonoma("04");
solicitud.setFechaConsulta("30/12/2021");
solicitud.setFechaNacimiento("17/08/1970");
ScspRespuesta respuesta =
    client.peticionSincrona(Arrays.asList(solicitud));
assertNotNull(respuesta);
}
```

### 3.2. Servei d'informes

Per a fer peticions a aquest servei s'ha de crear una instància de la següent classe:

```
es.caib.pinbal.client.informe.ClientInforme
```

Per a crear una instància d'aquest client s'ha d'utilitzar el següent codi:

```
ClientEstadistica client = new ClientInforme(
    URL_BASE,
    USUARI,
    CONTRASENYA);
```

Els paràmetres necessaris per a crear la instància del client son:

- URL\_BASE: l'adreça base de l'aplicació PINBAL (per exemple: <https://proves.caib.es/pinbalapi>).
- USUARI: el codi d'un usuari amb el rol PBL\_REPORT.
- CONTRASENYA: la contrasenya de l'usuari.

Els mètodes disponibles en aquest client son els següents (per a més detall podeu consultar el capítol 2.2 d'aquest document):

```
public List<Entitat> procedimientos();

public List<Entitat> usuarios();

public List<Servei> serveis();

public List<Entitat> general(
    Date dataInici,
    Date dataFi);
```

A continuació teniu un una classe d'exemple que mostra com fer peticions a aquest servei:

```

public class InformeTest {

    private static final String URL_BASE = "http://localhost:8080/pinbalapi";
    private static final String USUARI = "USER";
    private static final String CONTRASENYA = "PASSWD";

    private ClientInforme client = new ClientInforme(URL_BASE, USUARI,
CONTRASENYA);

    @Test
    public void procediments() {
        List<Entitat> resposta = client.procediments();
        assertNotNull(resposta);
    }

    @Test
    public void serveis() {
        List<Servei> resposta = client.serveis();
        assertNotNull(resposta);
    }

    @Test
    public void usuaris() {
        List<Entitat> resposta = client.usuaris();
        assertNotNull(resposta);
    }
}
  
```

### 3.3. Servei d'estadístiques

Per a fer peticions a aquest servei s'ha de crear una instància de la següent classe:

```
es.caib.pinbal.client.estadistica.ClientEstadistica
```

Per a crear una instància d'aquest client s'ha d'utilitzar el següent codi:

```

ClientEstadistica client = new ClientEstadistica(
    URL_BASE,
    USUARI,
    CONTRASENYA);
  
```

Els paràmetres necessaris per a crear la instància del client son:

- URL\_BASE: l'adreça base de l'aplicació PINBAL (per exemple: <https://proves.caib.es/pinbalapi>).
- USUARI: el codi d'un usuari amb el rol PBL\_REPORT.

- CONTRASENYA: la contrasenya de l'usuari.

Els mètodes disponibles en aquest client son els següents (per a més detall podeu consultar el capítol 2.2 d'aquest document):

```

public List<Procediment> consultes(
    String entitatCodi,
    String procedimentCodi,
    String serveiCodi,
    String estat,
    Date dataInici,
    Date dataFi);

public List<Entitat> carrega();
  
```

A continuació teniu un una classe d'exemple que mostra com fer peticions a aquest servei:

```

public class EstadisticaTest {

    private static final String URL_BASE = "http://localhost:8080/pinbalapi";
    private static final String USUARI = "USER";
    private static final String CONTRASENYA = "PASSWD";

    private ClientEstadistica client = new ClientEstadistica(URL_BASE,
    USUARI, CONTRASENYA);

    @Test
    public void carrega() throws IOException {
        List<Entitat> resposta = client.carrega();
        assertNotNull(resposta);
    }

}
  
```

### 3.4. Servei de dades obertes

Per a fer peticions a aquest servei s'ha de crear una instància de la següent classe:

```
es.caib.pinbal.client.dadesobertes.ClientDadesObertes
```

Per a crear una instància d'aquest client s'ha d'utilitzar el següent codi:

```

ClientEstadistica client = new ClientDadesObertes(
    URL_BASE,
    USUARI,
    CONTRASENYA);
  
```



Els paràmetres necessaris per a crear la instància del client son:

- URL\_BASE: l'adreça base de l'aplicació PINBAL (per exemple: <https://proves.caib.es/pinbalapi>).
- USUARI: el codi d'un usuari amb el rol PBL\_REPORT.
- CONTRASENYA: la contrasenya de l'usuari.

Els mètodes disponibles en aquest client son els següents (per a més detall podeu consultar el capítol 2.2 d'aquest document):

```

public List<DadesObertesRespostaConsulta>.opendata(
    String entitatCodi,
    Date dataInici,
    Date dataFi,
    String procedimentCodi,
    String serveiCodi);
  
```

A continuació teniu un una classe d'exemple que mostra com fer peticions a aquest servei:

```

public class InformeTest {

    private static final String URL_BASE = "http://localhost:8080/pinbalapi";
    private static final String USUARI = "USER";
    private static final String CONTRASENYA = "PASSWD";

    private ClientDadesObertes client = new ClientDadesObertes(URL_BASE,
        USUARI, CONTRASENYA);

    @Test
    public void.opendata() throws IOException {
        Calendar cal = Calendar.getInstance();
        cal.set(Calendar.DAY_OF_YEAR, 1);
        List<DadesObertesRespostaConsulta> resposta = client.opendata(
            "CAIB",
            cal.getTime(),
            new Date(),
            null,
            null);
        assertNotNull(resposta);
    }
}
  
```