



EIEL-ImpresionMapas

ESPECIFICACIÓN DEL SUBSISTEMA

Proyecto	EIEL-ImpresionMapas
Responsable	David Trillo Pérez
Versión	0.1
Fecha	Fecha de la última modificación
Estado	Final
Clasificación	Público

Índice

1. Introducción.....	3
2. Arquitectura global del sistema	15
Historial de Cambios	17

1. Introducción

Nombre del proyecto: [EIEL-ImpresionMapas](#)

EIEL-ImpresionMapas contiene utilidades varias para la impresión de cartografía.

Estructura de directorios:

- bin: Contiene los .class del subsistema
- dist: Contiene EIEL-ImpresionMapas.jar con las clases del subsistema.
- lib: Contiene las diferentes librerías utilizadas por el subsistema.
- src: Contiene los fuentes del subsistema.
- imaxesPDF: Contiene imágenes para la generación de los PDFs.
- leyendasImpresion: Contiene las imágenes correspondientes a las leyendas de los mapas de webEiel.
- javadoc: Contiene la documentación del módulo.

Dependencias con subsistemas:

- EIEL-Utilidades: [es.udc.lbd.eiel.util](#)

Estructura de paquetes:

- Directorio “base” del subsistema: [es.udc.lbd.eiel.impresionmapas](#).
- Estructura de directorios:
 - [es.udc.lbd.eiel.impresionmapas.document](#): Contiene la factoría que crea los documentos cartográficos a través de una clase abstracta que mediante el método plantilla proporciona una estructura de creación de documentos, así como una implementación por defecto.
 - FormatDocumentFactory
 - FormatDocument
 - FormatDocumentPDF
 - FormatDocumentRTF
 - [es.udc.lbd.eiel.impresionmapas.document.paint](#): Contiene una estructura de clases que siguiendo el patrón Estrategia proporcionan distintas formas de pintado para crear el documento cartográfico.
 - PaintStrategy: Clase abstracta que define los métodos abstractos que deben implementar las distintas subclases.
 - PaintMap: Subclase que proporciona un método que pinta los elementos de la cartografía sin más información que la extraída de la plantilla XML.
 - PaintSimpleSerie: pinta una serie cartográfica de forma sencilla, es decir dividiéndola en cuadros a lo largo del bbox, sin interseccionar con la geometría.
 - PaintMapsSeries: Clase que proporciona métodos generales a sus subclases para la creación de series cartográficas interseccionadas con la geometría del mapa solicitado.
 - PaintSerieWithGeometry: Genera una serie cartográfica teniendo en cuenta la escala y la geometría del mapa.

- [es.udc.lbd.eiel.impresionmapas.map.cache](#): Incluye una clase que hace caché de la información extraída del XML. Además implementa un thread que cada X tiempo (configurable en el fichero .properties) refresca dicha información por si ha sido modificada, de manera que no se necesite relanzar el programa
 - MapTemplates.java: clase que se encarga de refrescar la cache con los datos del XML cada cierto tiempo.
- [es.udc.lbd.eiel.impresionmapas.map.dao](#): Este paquete contiene una interfaz, una factoría y una implementación de la interfaz para poder acceder a los datos de manera que se oculte la tecnología empleada.
 - MapDAOFactory: devuelve el DAO apropiado para el acceso a la información cartográfica.
 - MapDAO: interfaz que define los métodos que permiten acceder a la información cartográfica.
 - StandardMapDAO: implementación de la interfaz que permite el acceso a la plantilla XML para así extraer la información.
- [es.udc.lbd.eiel.impresionmapas.map.dao.parser](#):
 - GMLParser: parser utilizado conjuntamente con el paquete geometry. Parsea GML. No se utiliza directamente para la impresión cartográfica.
 - Parser: clase que parsea y valida la plantilla XML con la información cartográfica.
- [es.udc.lbd.eiel.impresionmapas.map.dao.proxies](#):
 - ProxyWMS: el acceso al WMS se hace por medio de un Proxy ocultando así que se accede a una fuente de datos remota.
 - ProxyWFS: el acceso al WFS se hace por medio de un Proxy ocultando así que se accede a una fuente de datos remota. Este Proxy se utiliza junto con el paquete geometry, no directamente para generar cartografía.
- [es.udc.lbd.eiel.impresionmapas.map.dao.util](#):
 - ConventionTags: interfaz con constantes que hacen referencia a los tags de la plantilla XML de impresión.
 - WMSParserRequest: parsea la petición al WMS para devolverla en forma de String.
- [es.udc.lbd.eiel.impresionmapas.map.mapelements](#): Este paquete contiene objetos que siguiendo el patrón composición, representan los distintos elementos que se pueden pintar en la cartografía. Es fácil añadir y modificar estos objetos gracias a la estructura de diseño utilizada.
 - ConventionTags: interfaz con constantes que hacen referencia a los tags de la plantilla XML de impresión.
- [es.udc.lbd.eiel.impresionmapas.map.to](#): Este paquete contiene objetos que siguen el patrón (Transfer Object):

- MapSourceTO: información sobre la fuente de un mapa.
- MapTemplateTO: información sobre la plantilla de un mapa.
- ParameterTO: información sobre los parámetros (textos, imágenes, etc) de la cartografía.
- [es.udc.lbd.eiel.impresionmapas.map.util](#): paquete con utilidades para facilitar el manejo
 - BoundingBox: permite manipular la información sobre el área de un mapa (convertirla a un String, a un objeto BoundingBox, etc).
 - MapsSeries: permite a partir del bbox obtener el número áreas resultantes de dividirlo en x partes.
 - GlobalNames: clase con variables Strings que se utilizan en las demás clases.
 - XYConverter: clase que proporciona métodos para convertir pixeles en mm y viceversa.
 - Utilities: clase con métodos dispares para el manejo de ficheros y otras utilidades.
- [es.udc.lbd.eiel.impresionmapas.mapfacade](#): paquete con las clases que permiten el acceso a los casos de uso del módulo.
 - MapFacadeDelegateFactory: permite obtener una instancia de la fachada adecuado para acceder a las funcionalidades del módulo.
 - MapFacadeDelegate: interfaz con los casos de uso.
 - StandardMapFacadeDelegate: clase que implementa los casos de uso de la interfaz.
- [es.udc.lbd.eiel.impresionmapas.predefinedmaps](#): paquete con las clases que permiten generar los mapas predefinidos de la web.
 - MapaPredefinido: superclase que define los métodos comunes para la impresión de los mapas predefinidos.
 - MunicipioPredefinido: clase que permite obtener el mapa predefinido de un municipio.
 - NucleoPredefinido: clase que permite obtener el mapa predefinido de un núcleo.

JavaDoc

es.udc.lbd.eiel.impresionmapas.mapfacade

Class StandardMapFacadeDelegate

java.lang.Object

└ **es.udc.lbd.eiel.impresionmapas.mapfacade.StandardMapFacadeDelegate**

All Implemented Interfaces:

[MapFacadeDelegate](#)

```
public class StandardMapFacadeDelegate
  extends java.lang.Object
  implements MapFacadeDelegate
```

Constructor Summary

StandardMapFacadeDelegate ()

Method Summary

<u>MapElementComponent</u>	<u>getMapElementComponent</u> (int mapIdentifier, java.lang.String mapElementName, java.lang.String mapElementIdentifier) Entradas: - Identificador del mapa del que queremos obtener los elementos de la cartografía.
java.util.HashMap	<u>getMapImageParameter</u> () Salidas: - HashMap con los parametros de la cartografía que son imágenes.
java.util.HashMap	<u>getMapParameters</u> (int mapIdentifier) Entradas: - Identificador del mapa del que queremos obtener los parámetros.
java.util.HashMap	<u>getMapTemplatesList</u> () Salidas: - HashMap con los objetos MapTemplateTO creados a partir de la plantilla XML de impresión.
java.lang.String	<u>getPetitionWMSMunicipio</u> (java.lang.String urlWMS, java.lang.String municipio, java.lang.String minx, java.lang.String miny, java.lang.String maxx, java.lang.String maxy, int alto, int ancho, java.lang.String color) Entradas: - Url de servidor WMS que nos devuelve el mapa.
java.lang.String	<u>getPetitionWMSMunicipioTexto</u> (java.lang.String urlWMS, java.lang.String municipio, java.lang.String minx, java.lang.String miny, java.lang.String maxx, java.lang.String maxy, int alto, int ancho, java.lang.String color) Devuelve el nombre del municipio y el de los municipios adyacentes.
java.lang.String	<u>getPetitionWMSNucleo</u> (java.lang.String urlWMS, java.lang.String municipio, java.lang.String entidad, java.lang.String nucleo, java.lang.String minx, java.lang.String miny, java.lang.String maxx, java.lang.String maxy, int alto, int ancho, java.lang.String colorNucleoPrincipal, java.lang.String colorRestoNucleos) Entradas:

	- Url de servidor WMS que nos devuelve el mapa.
java.lang.String	<u>getPeticionWMSNucleoTexto</u> (java.lang.String urlWMS, java.lang.String municipio, java.lang.String entidad, java.lang.String nucleo, java.lang.String minx, java.lang.String miny, java.lang.String maxx, java.lang.String maxy, int alto, int ancho, java.lang.String colorNucleoPrincipal, java.lang.String colorRestoNucleos) Idem que el anterior, pero devuelve el texto de los núcleos.
java.util.Collection	<u>getTemplateMapSources</u> (int mapIdentifier) Entradas: - Identificador del mapa del que queremos las fuentes.
byte[]	<u>printMap</u> (int mapIdentifier, java.lang.String formatExtension) Entradas: - Identificador del mapa del que queremos imprimir.
byte[]	<u>printMap</u> (int mapIdentifier, java.lang.String formatExtension, java.util.Collection mapSources, java.util.HashMap parameters) Entradas: - Identificador del mapa del que queremos imprimir.
byte[]	<u>printMap</u> (int mapIdentifier, java.lang.String formatExtension, java.util.Collection mapSources, java.util.HashMap parameters, java.util.HashMap legend) Entradas: - Identificador del mapa del que queremos imprimir.
byte[]	<u>printMapsSeries</u> (int mapIdentifier, java.lang.String formatExtension, java.util.Collection geometries, java.util.Collection mapSources, java.util.HashMap parameters, double rectangleWidth, double rectangleHeight) Caso de uso que imprime a serie dun mapa.
byte[]	<u>printMapsSeries</u> (int mapIdentifier, java.lang.String formatExtension, java.util.Collection geometries, java.util.Collection mapSources, java.util.HashMap parameters, double rectangleWidth, double rectangleHeight, java.util.HashMap legend) Caso de uso que imprime a serie dun mapa.
byte[]	<u>printMapsSeries</u> (int mapIdentifier, java.lang.String formatExtension, java.lang.String layer, java.util.Collection mapSources, java.util.HashMap parameters, int mapsHeight,

	int mapsWidth) Caso de uso que imprime a serie dun mapa.
--	---

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

StandardMapFacadeDelegate

public StandardMapFacadeDelegate()

Method Detail

printMap

```
public byte[] printMap(int mapIdentifier,
                       java.lang.String formatExtension)
                       throws URLWMSEException,
```

es.udc.lbd.eiel.util.exceptions.InternalErrorException

Description copied from interface: [MapFacadeDelegate](#)

Entradas:

- Identificador del mapa del que queremos imprimir.

- Formato del documento a imprimir (PDF).

Salidas:

- Array de bytes del documento cartográfico.

Specified by:

[printMap](#) in interface [MapFacadeDelegate](#)

Throws:

[URLWMSEException](#)

es.udc.lbd.eiel.util.exceptions.InternalErrorException

printMap

```
public byte[] printMap(int mapIdentifier,
                       java.lang.String formatExtension,
                       java.util.Collection mapSources,
                       java.util.HashMap parameters)
                       throws
```

es.udc.lbd.eiel.util.exceptions.InternalErrorException,
[URLWMSEException](#)

Description copied from interface: [MapFacadeDelegate](#)

Entradas:

- Identificador del mapa del que queremos imprimir.
- Formato del documento a imprimir (PDF).
- Colección de MapSourceTO que han sido modificados..
- HashMap con los parámetros de impresión modificados ().

Salidas:

- Array de bytes del documento cartográfico.

Specified by:

[printMap](#) in interface [MapFacadeDelegate](#)

Throws:

[es.udc.lbd.eiel.util.exceptions.InternalErrorException](#)
[URLWMSEException](#)

printMap

```
public byte[] printMap(int mapIdentifier,  
                        java.lang.String formatExtension,  
                        java.util.Collection mapSources,  
                        java.util.HashMap parameters,  
                        java.util.HashMap legend)  
                        throws  
es.udc.lbd.eiel.util.exceptions.InternalErrorException,  
                        URLWMSEException
```

Description copied from interface: [MapFacadeDelegate](#)

Entradas:

- Identificador del mapa del que queremos imprimir.
- Formato del documento a imprimir (PDF).
- Colección de MapSourceTO que han sido modificados..
- HashMap con los parámetros de impresión modificados ().
- HashMap con los objetos Legend que representan las leyendas.

Salidas:

- Array de bytes del documento cartográfico.

Specified by:

[printMap](#) in interface [MapFacadeDelegate](#)

Parameters:

`mapIdentifier` - identificador do mapa
`formatExtension` - format document
`mapSources` - colección cas fontes do mapas que queremos imprimir
`parameters` - colección cos parámetros que queremos imprimir
`legend` - conxunto de lendas (texto, cor) que se queren imprimir

Returns:

byte[] bytes que forman o documento final.

Throws:

es.udc.lbd.eiel.util.exceptions.InternalErrorException
[URLWMSEException](#)

getMapTemplatesList

```
public java.util.HashMap getMapTemplatesList()  
                           throws  
es.udc.lbd.eiel.util.exceptions.InternalErrorException
```

Description copied from interface: [MapFacadeDelegate](#)

Salidas:

- HashMap con los objetos MapTemplateTO creados a partir de la plantilla XML de impresión.

Specified by:

[getMapTemplatesList](#) in interface [MapFacadeDelegate](#)

Throws:

es.udc.lbd.eiel.util.exceptions.InternalErrorException

getTemplateMapSources

```
public java.util.Collection getTemplateMapSources(int mapIdentifier)  
                           throws  
es.udc.lbd.eiel.util.exceptions.InternalErrorException
```

Description copied from interface: [MapFacadeDelegate](#)

Entradas:

- Identificador del mapa del que queremos las fuentes.

Salidas:

- Collection con los objetos MapSourceTO del mapa con el identificador mapIdentifier.

Specified by:

[getTemplateMapSources](#) in interface [MapFacadeDelegate](#)

Throws:

es.udc.lbd.eiel.util.exceptions.InternalErrorException

printMapsSeries

```
public byte[] printMapsSeries(int mapIdentifier,  
                               java.lang.String formatExtension,  
                               java.lang.String layer,  
                               java.util.Collection mapSources,  
                               java.util.HashMap parameters,  
                               int mapsHeight,  
                               int mapsWidth)  
                           throws URLWMSEException,
```

es.udc.lbd.eiel.util.exceptions.InternalErrorException

Caso de uso que imprime a serie dun mapa.

Specified by:

[printMapsSeries](#) in interface [MapFacadeDelegate](#)

Parameters:

mapIdentifier - identificador do mapa.

formatExtension - formato do documento

mapSources - fontes do mapa das que queremos obter a serie.

parameters - parametros (lendas, textos,etc) que van acompañar á serie no documento.

layer - capa da que se obterá a xeometría para xerar a serie.

Throws:

[URLWMSEException](#)

es.udc.lbd.eiel.util.exceptions.InternalErrorException

printMapsSeries

```
public byte[] printMapsSeries(int mapIdentifier,  
                               java.lang.String formatExtension,  
                               java.util.Collection geometries,  
                               java.util.Collection mapSources,  
                               java.util.HashMap parameters,  
                               double rectangleWidth,  
                               double rectangleHeight,  
                               java.util.HashMap legend)  
    throws URLWMSEException,
```

es.udc.lbd.eiel.util.exceptions.InternalErrorException

Caso de uso que imprime a serie dun mapa.

Specified by:

[printMapsSeries](#) in interface [MapFacadeDelegate](#)

Parameters:

mapIdentifier - identificador do mapa.

formatExtension - formato do documento

mapSources - fontes do mapa das que queremos obter a serie.

parameters - parametros (lendas, textos,etc) que van acompañar á serie no documento.

legends - lendas do mapa.

Throws:

[URLWMSEException](#)

es.udc.lbd.eiel.util.exceptions.InternalErrorException

printMapsSeries

```
public byte[] printMapsSeries(int mapIdentifier,  
                               java.lang.String formatExtension,  
                               java.util.Collection geometries,  
                               java.util.Collection mapSources,  
                               java.util.HashMap parameters,  
                               double rectangleWidth,  
                               double rectangleHeight)  
    throws URLWMSEException,
```

es.udc.lbd.eiel.util.exceptions.InternalErrorException

Caso de uso que imprime a serie dun mapa.

Specified by:

[printMapsSeries](#) in interface [MapFacadeDelegate](#)

Parameters:

mapIdentifier - identificador do mapa.

formatExtension - formato do documento

mapSources - fontes do mapa das que queremos obter a serie.

parameters - parametros (lendas, textos,etc) que van acompañar á serie no documento.

layer - capa da que se obterá a xeometría para xerar a serie.

Throws:

[URLWMSEException](#)

es.udc.lbd.eiel.util.exceptions.InternalErrorException

getMapParameters

```
public java.util.HashMap getMapParameters(int mapIdentifier)
                                throws
es.udc.lbd.eiel.util.exceptions.InternalErrorException
```

Description copied from interface: [MapFacadeDelegate](#)

Entradas:

- Identificador del mapa del que queremos obtener los parámetros.

Salidas:

- HashMap con los parametros.

Specified by:

[getMapParameters](#) in interface [MapFacadeDelegate](#)

Throws:

es.udc.lbd.eiel.util.exceptions.InternalErrorException

getMapElementComponent

```
public MapElementComponent getMapElementComponent(int mapIdentifier,
java.lang.String mapElementName,
java.lang.String mapElementIdentifier)
                                throws
es.udc.lbd.eiel.util.exceptions.InternalErrorException
```

Description copied from interface: [MapFacadeDelegate](#)

Entradas:

- Identificador del mapa del que queremos obtener los elementos de la cartografía.

- Tipo del elemento.

- Identificador del elemento.

Salidas:

- Objeto MapElementComponent del que podremos obtener la información deseada..

Specified by:

[getMapElementComponent](#) in interface [MapFacadeDelegate](#)

Throws:

es.udc.lbd.eiel.util.exceptions.InternalErrorException

getPeticiónWMSMunicipio

```
public java.lang.String
getPeticiónWMSMunicipio(java.lang.String urlWMS,

java.lang.String municipio,

                                java.lang.String minx,
                                java.lang.String miny,
                                java.lang.String maxx,
                                java.lang.String maxy,
                                int alto,
                                int ancho,

java.lang.String color)
```

Description copied from interface: [MapFacadeDelegate](#)

Entradas:

- Url de servidor WMS que nos devuelve el mapa.
- Código del municipio deseado.
- Bbox do municipio -> minx, miny, maxx, maxy.
- Ancho y alto del mapa.
- Color en el que será devuelto el mapa.

Salidas:

- Petición que se le debe hacer al servidor WMS (tiene body sld)..

Specified by:

[getPeticiónWMSMunicipio](#) in interface [MapFacadeDelegate](#)

getPeticiónWMSMunicipioTexto

```
public java.lang.String
getPeticiónWMSMunicipioTexto(java.lang.String urlWMS,

java.lang.String municipio,

java.lang.String minx,

java.lang.String miny,

java.lang.String maxx,

java.lang.String maxy,

                                int alto,
                                int ancho,

java.lang.String color)
```

Description copied from interface: [MapFacadeDelegate](#)

Devuelve el nombre del municipio y el de los municipios adyacentes.

Entradas:

- Url de servidor WMS que nos devuelve el mapa.
- Código del municipio deseado.
- Bbox do municipio -> minx, miny, maxx, maxy.
- Ancho y alto del mapa.
- Color en el que será devuelto el mapa.

Salidas:

- Petición que se le debe hacer al servidor WMS (tiene body sld)..

Specified by:

[getPeticionWMSMunicipioTexto](#) in interface [MapFacadeDelegate](#)

getPeticionWMSNucleo

```
public java.lang.String getPeticionWMSNucleo(java.lang.String urlWMS,  
  
java.lang.String municipio,  
  
java.lang.String entidad,  
java.lang.String nucleo,  
java.lang.String minx,  
java.lang.String miny,  
java.lang.String maxx,  
java.lang.String maxy,  
int alto,  
int ancho,  
  
java.lang.String colorNucleoPrincipal,  
java.lang.String colorRestoNucleos)
```

Description copied from interface: [MapFacadeDelegate](#)

Entradas:

- Url de servidor WMS que nos devuelve el mapa.
- Código del municipio deseado.
- Código de la entidad del nucleo.
- Código del núcleo.
- Bbox do núcleo -> minx, miny, maxx, maxy.
- Ancho y alto del mapa.
- Color en el que será devuelto el núcleo.
- Color en el que serán devueltos el resto de núcleos.

Salidas:

- Petición que se le debe hacer al servidor WMS (tiene body sld)..

Specified by:

[getPeticionWMSNucleo](#) in interface [MapFacadeDelegate](#)

getPeticionWMSNucleoTexto

```
public java.lang.String  
getPeticionWMSNucleoTexto(java.lang.String urlWMS,  
  
java.lang.String municipio,  
  
java.lang.String entidad,  
  
java.lang.String nucleo,  
  
java.lang.String minx,  
  
java.lang.String miny,  
  
java.lang.String maxx,  
  
java.lang.String maxy,  
  
int alto,
```

```
int ancho,  
  
java.lang.String colorNucleoPrincipal,  
  
java.lang.String colorRestoNucleos)
```

Description copied from interface: [MapFacadeDelegate](#)

Idem que el anterior, pero devuelve el texto de los núcleos.

Entradas:

- Url de servidor WMS que nos devuelve el mapa.
- Código del municipio deseado.
- Código de la entidad del nucleo.
- Código del núcleo.
- Bbox do núcleo -> minx, miny, maxx, maxy.
- Ancho y alto del mapa.
- Color en el que será devuelto el núcleo.
- Color en el que serán devueltos el resto de núcleos.

Salidas:

- Petición que se le debe hacer al servidor WMS (tiene body sld)..

Specified by:

[getPeticionWMSNucleoTexto](#) in interface [MapFacadeDelegate](#)

getMapImageParameter

```
public java.util.HashMap getMapImageParameter( )  
                                throws  
es.udc.lbd.eiel.util.exceptions.InternalErrorException
```

Description copied from interface: [MapFacadeDelegate](#)

Salidas:

- HashMap con los parametros de la cartografía que son imágenes.

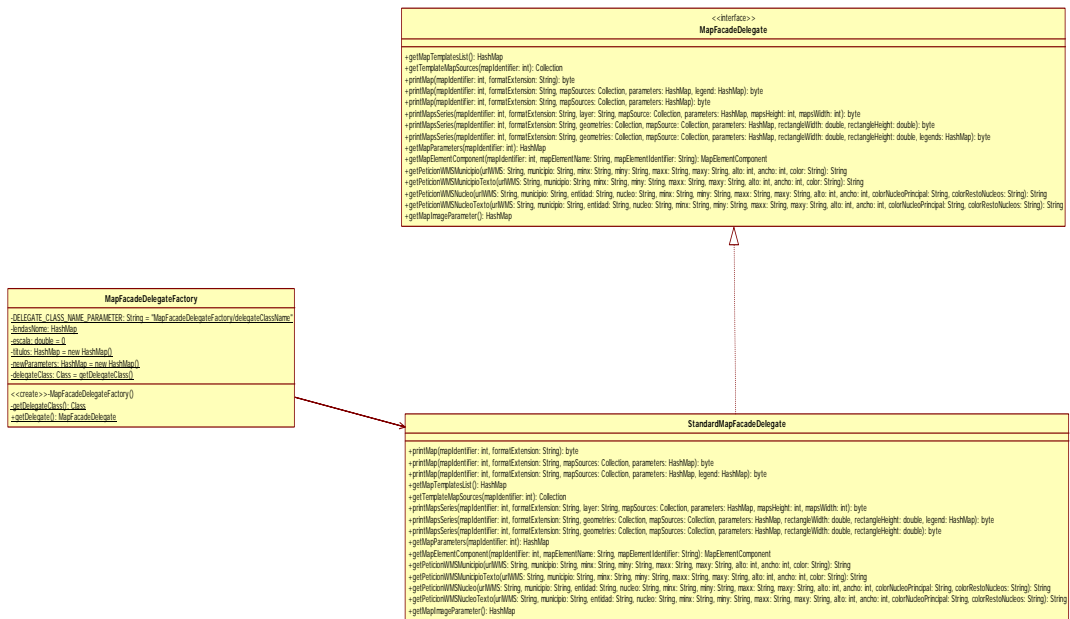
Specified by:

[getMapImageParameter](#) in interface [MapFacadeDelegate](#)

Throws:

es.udc.lbd.eiel.util.exceptions.InternalErrorException

2. Arquitectura global del sistema



Historial de Cambios

Versión	Fecha	Cambios
0.1	21/06/2007	Versión Inicial