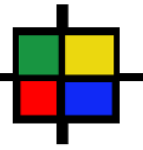


**calleÇpaña**



## **BASE DE DATOS DE CALLEÇPAÑA**

Autor: Carlos Javier Martín González.

Licenciado en Física Teórica por la Universidad Autónoma de Madrid.

Analista programador y funcional. Desarrollador independiente.

Revisión 1.0: 2011 - 04 - 22 (CalleÇpaña versión 1.0).

Revisión 1.1: 2012 - 02 - 08.

CalleÇpaña ® es una marca registrada.

Copyright © 2009 CalleÇpaña. Todos los derechos reservados.

# BASE DE DATOS DE CALLEÇPAÑA

## MANUAL DE ADMINISTRADOR

En este documento se describen las funcionalidades y características principales de la Base de Datos de CalleÇpaña. Todas las aplicaciones para usuarios finales del Sistema de Normalización de Direcciones de CalleÇpaña utilizan esta base de datos.

Personal a quien está dirigido este manual:

- Para todos los tipos de servidor (lectura necesaria):
  - Administradores del servidor de CalleÇpaña.
  - Perfiles funcionales encargados de la integración y la explotación de las aplicaciones de CalleÇpaña en los sistemas donde se utilice.
- Para los servidores tipo "Normalizador PL/SQL" (lectura recomendada):
  - Desarrolladores que utilicen la API PL/SQL de la base de datos para acceder al Normalizador de Direcciones desde sistemas que integren CalleÇpaña.

## ÍNDICE DE CONTENIDOS

- 1. [Características y Objetivos de la Base de Datos](#)
- 2. [Estructura de la Base de Datos](#)
- 3. [API PL/SQL de Acceso a la Base de Datos](#)
  - 3.1. [Características Técnicas de la API PL/SQL](#)
  - 3.2. [Características Funcionales del Callejero y el Normalizador de Direcciones](#)
    - 3.2.1. [Fuentes de Datos](#)
    - 3.2.2. [Referenciación Territorial](#)
    - 3.2.3. [Normalización de Direcciones, Códigos Postales y Nombres](#)
  - 3.3. [Paquete PL/SQL del Normalizador de Direcciones con Acceso Personalizado](#)

## 1 - CARACTERÍSTICAS Y OBJETIVOS DE LA BASE DE DATOS

Características técnicas de la Base de Datos:

- Tipo: Relacional, normalizada 4FN.
- Sistema de acceso: Informático.
- Modo de acceso: Indexado.
- Versión: 1.0.

Objetivos de la Base de Datos:

- Consulta del Callejero:
  - Gestionar los datos de un Callejero referencial, estadístico y postal de España basándose en lo referente a la información del callejero de la “*ORDEN de 11 de julio de 1997 sobre comunicaciones electrónicas entre las Administraciones públicas referentes a la información de los Padrones municipales*” publicada en el [BOE número 169](#), y de la “*RESOLUCIÓN de 9 de abril de 1997 por la que se dictan instrucciones técnicas a los ayuntamientos sobre la gestión y revisión del Padrón municipal*” publicada en el [BOE número 87](#), incluyendo también la posición geográfica a nivel de población y código postal.
  - Permitir acceder a todos los datos no obsoletos del Callejero de la Base de Datos, incluidos los datos erróneos en los ficheros del INE de acuerdo a la normativa vigente desde el año 1997, y proporcionando la referenciación territorial completa de todas las entidades territoriales consultadas.
- Normalización de Direcciones:
  - Gestionar y normalizar direcciones referenciales, estadísticas y postales de España con sus posiciones geográficas utilizando los datos del Callejero de la Base de Datos.
  - Buscar las entidades territoriales no obsoletas del Callejero de la Base de Datos, sin incluir las entidades duplicadas en los ficheros del INE de acuerdo a la normativa vigente desde el año 1997, para crear direcciones normalizadas dentro de ellas, y proporcionar la referenciación territorial de las entidades territoriales limpiando los nombres redundantes y los núcleos de población diseminados que no aportan ninguna información relevante.
  - Obtener todos los datos referenciales, estadísticos y postales, la geoposición, el nivel de normalización de la dirección y de su código postal, el estado de actualización y la duplicidad en el código de vía de las direcciones creadas y contenidas en el Normalizador de Direcciones.
  - Proporcionar todos los datos necesarios para que las direcciones normalizadas obtenidas del Normalizador puedan cumplir con la “*ORDEN EHA/3300/2008, de 7 de noviembre, por la que se aprueba el modelo 196, del Impuesto sobre la Renta de las Personas Físicas, Impuesto sobre Sociedades e Impuesto sobre la Renta de no Residentes (establecimientos permanentes). Retenciones e ingresos a cuenta sobre rendimientos del capital mobiliario y rentas obtenidos por la contraprestación derivada de cuentas en toda clase de instituciones financieras, incluyendo las basadas en operaciones sobre activos financieros, declaración informativa anual de personas autorizadas y de saldos en cuentas de toda clase de instituciones financieras.*”, publicada en el [BOE número 278](#).

- Administrar el Servidor (máquina virtual de CalleÇpaÑa):
  - Configurar el funcionamiento de las tres aplicaciones que utilizan la Base de Datos:
    - Callejero.
    - Normalizador de Direcciones (versiones Web y PL/SQL).
    - Administración del Servidor.
  - En los servidores con Normalizador de Direcciones (versiones Web y PL/SQL):
    - Consultar el estado de la Base de Datos y las direcciones que contiene.
    - Eliminar todos los datos obsoletos y liberar todo el espacio no utilizado en la Base de Datos para dejar espacio libre para nuevas direcciones.
  - En los servidores con Normalizador de Direcciones (versión PL/SQL):
    - Cambiar la contraseña del usuario de la Base de Datos con acceso al paquete de la API PL/SQL del Normalizador de Direcciones Personalizado.

## 2 - ESTRUCTURA DE LA BASE DE DATOS

La estructura de tablas y campos se muestra en el siguiente diagrama del modelo relacional de la base de datos. Los índices y los campos nulos no se muestran.

No se proporciona ninguna relación ni descripción de las tablas y campos de la base de datos porque el único modo de acceso a la misma es la "API PL/SQL del Normalizador de Direcciones con Acceso Personalizado".

Las referencias externas (FK) se pueden seguir teniendo en cuenta que los nombres de los campos que son clave única en la tabla "padre" de cada referencia externa se conserva en la clave foránea de la tabla "hija" dependiente.

Base de Datos de CalleCpaña - Tablas, Campos y Estructura

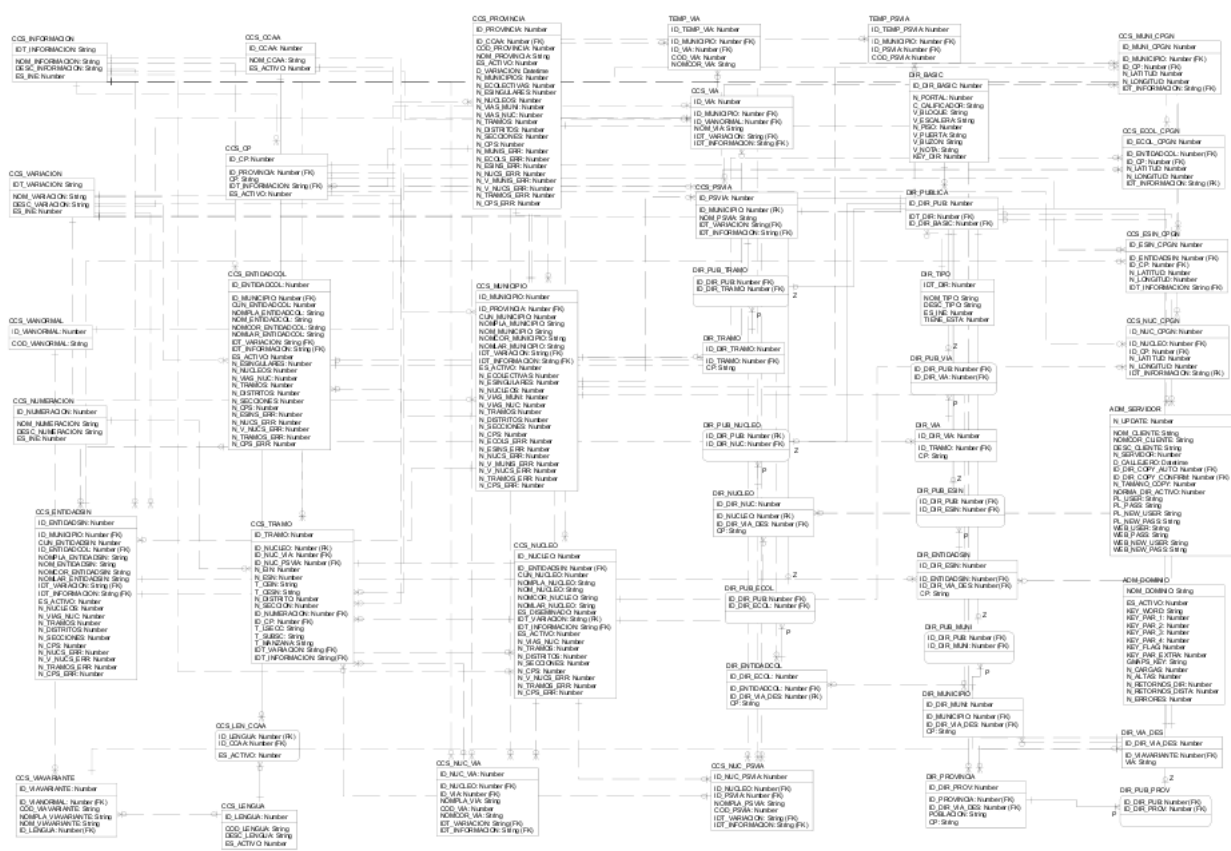


Diagrama de la Base de Datos de CalleCpaña sin índices ni campos nulos.

### 3 - API PL/SQL DE ACCESO A LA BASE DE DATOS

La API PL/SQL de acceso a la Base de Datos de CalleCpaña está compuesta por cuatro paquetes.

Solamente se proporciona acceso al paquete del Normalizador de Direcciones con Acceso Personalizado en los servidores de CalleCpaña del tipo "Normalizador PL/SQL".

El acceso al paquete del Normalizador con Acceso Personalizado se realiza mediante un sinónimo del nombre del paquete que puede variar en cada implementación de la Base de Datos.

La manera de acceder a los métodos de la API PL/SQL es:

```
sinonimo_del_paquete.nombre_de_funcion_o_procedimiento
```

Paquetes PL/SQL de acceso a la Base de Datos:

- Administración.
- Callejero.
- Normalizador de Direcciones con Acceso Anónimo (sin identificación en el acceso).
- Normalizador de Direcciones con Acceso Personalizado (con identificación en el acceso):

Objetivos del paquete PL/SQL del Normalizador de Direcciones con Acceso Personalizado:

- Consultar las "Claves de Seguridad de CalleCpaña" configuradas para los dominios Web.
- Consultar los datos identificativos del servidor y la fecha de los datos del Callejero que contiene.
- Consultar la clave única o el identificador de cliente de la API de Google Maps para los dominios Web.
- Comprobar la validez del nombre y la contraseña del usuario administrador del servidor.
- Buscar entidades territoriales no obsoletas del Callejero.
- Insertar direcciones en entidades territoriales no obsoletas del Callejero.
- Consultar todas las direcciones insertadas.

### 3.1 - CARACTERÍSTICAS TÉCNICAS DE LA API PL/SQL

#### Métodos de Acceso:

- FUNCTION: Función, retorna un parámetro aparte de los de salida.
- PROCEDURE: Procedimiento, no retorna ningún parámetro más que los de salida.

#### Sentido de los Parámetros:

- in: Parámetro de entrada.
- out: Parámetro de salida.
- in out: Parámetro de entrada y salida.

#### Tipos de Parámetros:

- INTEGER: Parámetro tipo número entero.
- NUMBER: Parámetro tipo número decimal.
- VARCHAR2: Parámetro tipo cadena.
- REF CURSOR: Parámetro tipo cursor.

#### Prefijos de los Nombres de los Parámetros (indican su utilidad):

- pNom: Nombre.
- pNum: Número.
- pFecha: Fecha.
- pCurr: Usuario o contraseña actual.
- pNivel: Nivel de normalización.
- pCod: Código.
- pGet: Booleano (Si / No).
- pHay: Booleano (Si / No).
- pMin: Valor mínimo.
- pMax: Valor máximo.
- pId: Identificador.
- pPar: Parámetro del sistema de seguridad de CalleÇpaÑa.
- pPos: Posición.
- pTipo: Tipo.
- pGeo: Posición geográfica.

### Nombres de Parámetros sin Prefijo:

- pPortal: Número de portal.
- pCalif: Calificador del número de portal.
- pBloque: Bloque.
- pEscale: Escalera.
- pPuerta: Puerta.
- pPiso: Piso.
- pBuzon: Buzón.
- pNota: Nota aclaratoria.
- pCP: Código postal.
- pDominio: Dominio Web.
- pWord: Palabra clave de paso del sistema de seguridad de CalleÇpaÑa.
- pFlag: Flag del sistema de seguridad de CalleÇpaÑa.
- pGMaps: Clave única o identificador de cliente de la API de Google Maps.

### Parámetros con Valores Predeterminados:

- pNivelDir (nivel de normalización de la dirección, descrito en este documento):
  - Valores posibles: todos del 0 al 12 inclusive.
- pNivelCP (nivel de normalización del código postal, descrito en este documento):
  - Valores posibles: -1, 0, 2, 4, 6, 8, 10, 12 y 13.
- pTipoPob (tipo de población):
  - 0: Núcleo de población.
  - 1: Entidad Singular.
  - 2: Entidad Colectiva.
  - 3: Municipio.
- Lado para pEscale (escalera) y pPuerta (puerta):
  - -1: Izquierda.
  - -2: Centro.
  - -3: Derecha.
- pPiso (número de piso):
  - = 0: Bajo.
  - < 0: Sótanos.
  - -100: No hay piso.
- pGeoLati (latitud geográfica):
  - 0: Posición geográfica desconocida.



### 3.2 - CARACTERÍSTICAS FUNCIONALES DEL CALLEJERO Y EL NORMALIZADOR DE DIRECCIONES

A continuación se detallan las características funcionales más importantes del "Callejero" y el "Normalizador de Direcciones" de la Base de Datos.

#### 3.2.1 - FUENTES DE DATOS

Las fuentes de datos que utilizan el "Callejero" y el "Normalizador de Direcciones" son:

- Ficheros de dominio público del "Callejero del Censo Electoral" de España publicados por el INE (<http://www.ine.es/>).
- Fichero con licencia "Creative Commons Attributions" de códigos postales y geoposiciones de España de GeoNames (<http://download.geonames.org/export/>).

El paquete de la API PL/SQL del "Callejero" permite acceder a todos los datos no obsoletos del Callejero, incluidos los datos erróneos en los ficheros del INE de acuerdo a la normativa vigente desde el año 1997.

Los dos paquetes del "Normalizador de Direcciones" permiten crear direcciones en entidades territoriales no obsoletas del Callejero, siempre que no correspondan a un dato duplicado de acuerdo a la normativa vigente desde el año 1997.

#### 3.2.2 - REFERENCIACIÓN TERRITORIAL

La referenciación territorial siempre se tiene en cuenta para todas las entidades territoriales, tanto en el "Callejero" como en el "Normalizador de Direcciones". El grado de referenciación de las entidades territoriales aumenta desde cada tipo de entidad "padre" (contenedora) hacia sus entidades "hijas" (contenidas).

Tipos de entidades territoriales del Callejero ordenadas de menor a mayor grado de referenciación:

- Comunidades Autónomas.
- Provincias.
- Municipios.
- Entidades Colectivas.
- Entidades Singulares.
- Núcleos de Población.
- Vías y Pseudovías.
- Tramos de vía o pseudovía.

El "lugar" es un tipo lógico de entidad territorial que comprende las entidades singulares, los núcleos de población y las pseudovías.

Las entidades territoriales devueltas por el "Callejero" contienen los nombres de todas las entidades referenciadas en la entidad devuelta, el "Normalizador de Direcciones" realiza una limpieza de los nombres redundantes y los núcleos de población diseminados que no aportan ninguna información relevante.

### 3.2.3 - NORMALIZACIÓN DE DIRECCIONES, CÓDIGOS POSTALES Y NOMBRES

En la Base de Datos hay un único "Normalizador de Direcciones" que permite dos tipos de acceso, uno anónimo y otro personalizado. Las direcciones dadas de alta en la Base de Datos se identifican con un identificador numérico único. La normalización de las direcciones en la Base de Datos asegura que cada identificador corresponda a una dirección única en la Base de Datos, y que todos los métodos de la API PL/SQL retornen el identificador de la dirección que ya existe en la Base de Datos cuando se intenta crear una dirección igual que otra que ya esté creada, de modo que no se puedan duplicar direcciones.

#### NIVEL DE NORMALIZACIÓN DE LA DIRECCIÓN

El nivel de normalización de una dirección se indica en el parámetro pNivelDir y es un indicador de la calidad y veracidad de la dirección según los datos del Callejero de acuerdo a la entidad territorial sobre la que se asienta la dirección y su información estadística (censal). Para su determinación se tiene en cuenta la entidad territorial normalizada en la dirección y, por orden de importancia:

- El grado de referenciación territorial de la entidad territorial normalizada, teniendo en cuenta el número de núcleos de población contenidos dentro de la misma en el caso de tratarse de una población del tipo municipio, entidad colectiva o entidad singular.
- La ambigüedad en la información estadística del número de portal de la dirección cuando la entidad territorial normalizada es un tramo de vía o pseudovía, y la homogeneidad en la información estadística cuando es cualquier otra entidad territorial.

El nivel de normalización de una dirección disminuye a medida que aumentan el grado de referenciación territorial y la homogeneidad de la información estadística para la entidad territorial normalizada de la dirección. El nivel de normalización aumenta si hay ambigüedad en la información estadística. La calidad y normalización de una dirección aumentan a medida que disminuye su nivel de normalización. Los niveles de normalización pares, a excepción del nivel 12 que corresponde a las direcciones desnormalizadas, indican homogeneidad en la información estadística de la entidad territorial sobre la que se asienta la dirección.

La información estadística de un número de portal de una dirección normalizada a nivel de tramo de vía o pseudovía se considera ambigua si el número de portal pertenece a más de un tramo en el Callejero con distinta información estadística, y se considera que la información estadística de una entidad territorial es homogénea si únicamente contiene un distrito, sección, letra de sección y subsección en toda la entidad.

Si una dirección está normalizada a nivel de población de los tipos municipio, entidad colectiva o entidad singular, y la población normalizada solamente contiene un núcleo de población, el nivel de normalización disminuye al nivel equivalente del núcleo de población.

Este método permite al "Normalizador de Direcciones" de CalleCpaña poder determinar trece niveles de normalización diferentes, desde el 0 para las direcciones mejor normalizadas, hasta el 12 para las direcciones desnormalizadas.

Niveles de Normalización de las direcciones según el grado de referenciación territorial y la información estadística (censal) de su entidad territorial normalizada:

- 0: Tramo de Vía o Pseudovía sin ambigüedad en la información estadística del número de portal de la dirección.
- 1: Tramo de Vía o Pseudovía con ambigüedad en la información estadística del número de portal de la dirección.
- 2: Vía o Pseudovía con homogeneidad en la información estadística del núcleo de población que la contiene.
- 3: Vía o Pseudovía sin homogeneidad en la información estadística del núcleo de población que la contiene.
- 4: Núcleo de Población con homogeneidad en su información estadística.
- 5: Núcleo de Población sin homogeneidad en su información estadística.
- 6: Entidad Singular con homogeneidad en su información estadística.
- 7: Entidad Singular sin homogeneidad en su información estadística.
- 8: Entidad Colectiva con homogeneidad en su información estadística.
- 9: Entidad Colectiva sin homogeneidad en su información estadística.
- 10: Municipio con homogeneidad en su información estadística.
- 11: Municipio sin homogeneidad en su información estadística.
- 12: Provincia:
  - No puede tener información estadística.
  - Corresponde a las direcciones desnormalizadas.

#### NIVEL DE NORMALIZACIÓN DEL CÓDIGO POSTAL

El nivel de normalización del código postal de una dirección se retorna en el parámetro pNivelICP y es un indicador de la calidad y veracidad del código postal de la dirección según los datos del Callejero de acuerdo a la entidad territorial sobre la que se asienta la dirección y sus códigos postales.

El Normalizador de Direcciones de CalleCpaÑa permite determinar nueve niveles diferentes de normalización del código postal de una dirección según el grado de coincidencia con los datos del Callejero. La calidad del código postal de una dirección es mejor cuanto menor sea su nivel normalización, siempre que el nivel no sea negativo.

Niveles de Normalización del Código Postal:

- -1: El código postal no existe para el Callejero.
- 0: El código postal coincide con el Callejero a nivel de Tramo.
- 2: El código postal coincide con el Callejero a nivel de Vía o Pseudovía.
- 4: El código postal coincide con el Callejero a nivel de Núcleo de Población.
- 6: El código postal coincide con el Callejero a nivel de Entidad Singular.
- 8: El código postal coincide con el Callejero a nivel de Entidad Colectiva.
- 10: El código postal coincide con el Callejero a nivel de Municipio.
- 12: El código postal coincide con el Callejero a nivel de Provincia.
- 13: El código postal no pertenece a la provincia (código postal desnormalizado).

## NORMALIZACIÓN DE NOMBRES

Al crear una dirección normalizada a nivel de población hay que introducir obligatoriamente el nombre de la vía o lugar al que pertenece la dirección y, opcionalmente, el tipo de vía. Cuando se crea una dirección desnormalizada también hay que introducir obligatoriamente el nombre del municipio al que pertenece la dirección.

Antes de crear una dirección en una vía desnormalizada o un municipio desnormalizado, el "Normalizador de Direcciones" realiza una búsqueda en el Callejero del nombre de la vía o lugar y el tipo de vía entre las vías y pseudovías normalizadas de su municipio, y del nombre del municipio entre las poblaciones normalizadas de su provincia. En el caso de encontrar alguna coincidencia, el "Normalizador de Direcciones" asignará automáticamente la vía o población normalizada del Callejero a la vía o municipio desnormalizado de la dirección que se está creando, y reducirá el nivel de normalización de acuerdo a la vía o población normalizadas coincidentes.

### 3.3 - PAQUETE DEL NORMALIZADOR DE DIRECCIONES CON ACCESO PERSONALIZADO

Con este paquete de la API PL/SQL se accede de manera personalizada al "Normalizador de Direcciones" con control de acceso. Se requiere la identificación por parámetro del nombre del cliente-propietario del servidor y el número de servidor en todas las funciones y procedimientos salvo en la función que prueba la conectividad con la base de datos sin control de acceso.

```
/* Retorna 1. Para probar la conectividad con la base de datos sin control de acceso. */
```

```
FUNCTION getNumberOne RETURN INTEGER;
```

```
/* Retorna 1. Para probar la conectividad con la base de datos. */
```

```
FUNCTION getNumberOne (pNomCliPro in VARCHAR2) RETURN INTEGER;
```

```
/* Retorna información sobre el cliente y la fecha de los datos del Callejero del servidor. */
```

```
PROCEDURE getBasicInfoServerLastUpdate (pNomCliPro in VARCHAR2 , pNomCli out VARCHAR2,  
pNomCorCli out VARCHAR2, pNumServ out INTEGER, pFechaCalle out DATE);
```

```
/* Comprueba si está activo el normalizador de direcciones. */
```

```
FUNCTION esNormalizadorDirsActivo (pNomCliPro in VARCHAR2) RETURN INTEGER;
```

```
/* Busca poblaciones y vías por su nombre. */
```

```
FUNCTION searchPobsVias (pNomCliPro in VARCHAR2,  
pCodPro in out INTEGER, pNomPob in out VARCHAR2,  
pPosPob in out INTEGER, pNomVia in out VARCHAR2, pPosVia in out INTEGER,  
pPosIniPag in out INTEGER, pNumRegPag in out INTEGER, pCodLengua in out INTEGER,  
pNumRegTot out INTEGER, pTipoCursor out INTEGER, pHayEntCol out INTEGER,  
pHayEntSin out INTEGER, pHayNucleo out INTEGER) RETURN REF CURSOR;
```

```
/* Busca vías por nombre de población y vía. */
```

```
FUNCTION searchVias (pNomCliPro in VARCHAR2,  
pCodPro in out INTEGER, pNomPob in out VARCHAR2,  
pPosPob in out INTEGER, pNomVia in out VARCHAR2, pPosVia in out INTEGER,  
pPosIniPag in out INTEGER, pNumRegPag in out INTEGER, pCodLengua in out INTEGER,  
pNumRegTot out INTEGER, pTipoCursor out INTEGER, pHayEntCol out INTEGER,  
pHayEntSin out INTEGER, pHayNucleo out INTEGER) RETURN REF CURSOR;
```

```
/* Busca poblaciones por su nombre. */
```

```
FUNCTION searchPobs (pNomCliPro in VARCHAR2,  
pCodPro in out INTEGER, pNomPob in out VARCHAR2,  
pPosPob in out INTEGER, pPosIniPag in out INTEGER, pNumRegPag in out INTEGER,  
pCodLengua in out INTEGER, pNumRegTot out INTEGER, pTipoCursor out INTEGER,  
pHayEntCol out INTEGER, pHayEntSin out INTEGER, pHayNucleo out INTEGER)  
RETURN REF CURSOR;
```

```
/* Retorna los datos de población y vía indicados. */
```

```
FUNCTION getPobVia (pNomCliPro in VARCHAR2,  
pCodPro in out INTEGER, pNomPob in out VARCHAR2,  
pIdNucVia in out INTEGER, pNomVia in out VARCHAR2, pIdNucPsVia in out INTEGER,  
pPosIniPag in out INTEGER, pNumRegPag in out INTEGER, pCodLengua in out INTEGER,  
pNumRegTot out INTEGER, pTipoCursor out INTEGER, pHayEntCol out INTEGER,  
pHayEntSin out INTEGER, pHayNucleo out INTEGER, pCP out VARCHAR2)  
RETURN REF CURSOR;
```

```

/* Retorna los datos de población indicada. */
FUNCTION getPoblacion (pNomCliPro in VARCHAR2,
    pCodPro in out INTEGER, pTipoPob in INTEGER, pIdPob in INTEGER,
    pNomPob in out VARCHAR2, pIdNucVia in out INTEGER, pNomVia in out VARCHAR2,
    pIdNucPsVia in out INTEGER, pPosIniPag in out INTEGER, pNumRegPag in out INTEGER,
    pCodLengua in out INTEGER, pNumRegTot out INTEGER, pTipoCursor out INTEGER,
    pHayEntCol out INTEGER, pHayEntSin out INTEGER, pHayNucleo out INTEGER,
    pCP out VARCHAR2) RETURN REF CURSOR;

/* Retorna la provincia indicada o todas si no se indica. */
FUNCTION getProvincias (pNomCliPro in VARCHAR2, pCodProv in INTEGER)
    RETURN REF CURSOR;

/* Comprueba la dirección normalizada a nivel de vía o tramo de vía. */
PROCEDURE checkDirViaNor (pNomCliPro in VARCHAR2, pGetNivelCP in INTEGER,
    pIdNVia in INTEGER, pIdNPsVia in INTEGER, pCodPro in out INTEGER,
    pCP in out VARCHAR2, pPortal in out INTEGER, pCalif in out VARCHAR2,
    pBloque in out VARCHAR2, pEscale in out VARCHAR2, pPiso in out INTEGER,
    pPuerta in out VARCHAR2, pBuzon in out VARCHAR2, pNota in out VARCHAR2,
    pNivelDir out INTEGER, pNomNivel out VARCHAR2, pNomPro out VARCHAR2,
    pNomMuni out VARCHAR2, pNomECol out VARCHAR2, pNomESin out VARCHAR2,
    pNomNucleo out VARCHAR2, pTipoVia out VARCHAR2, pNomVia out VARCHAR2,
    pDistrito out INTEGER, pSeccion out INTEGER, pLetraSecc out VARCHAR2,
    pSubSecc out VARCHAR2, pNivelCP out INTEGER);

/* Crea si no existe y retorna el identificador de la dirección normalizada a nivel de vía o tramo de vía. */
FUNCTION getIdDirViaNor (pNomCliPro in VARCHAR2, pGetNivelCP in INTEGER,
    pIdNVia in INTEGER, pIdNPsVia in INTEGER, pCodPro in out INTEGER,
    pCP in out VARCHAR2, pPortal in out INTEGER, pCalif in out VARCHAR2,
    pBloque in out VARCHAR2, pEscale in out VARCHAR2, pPiso in out INTEGER,
    pPuerta in out VARCHAR2, pBuzon in out VARCHAR2, pNota in out VARCHAR2,
    pNivelDir out INTEGER, pNomNivel out VARCHAR2, pNomPro out VARCHAR2,
    pNomMuni out VARCHAR2, pNomECol out VARCHAR2, pNomESin out VARCHAR2,
    pNomNucleo out VARCHAR2, pTipoVia out VARCHAR2, pNomVia out VARCHAR2,
    pDistrito out INTEGER, pSeccion out INTEGER, pLetraSecc out VARCHAR2,
    pSubSecc out VARCHAR2, pNivelCP out INTEGER) RETURN INTEGER;

/* Retorna los tipos de vía de la provincia indicada o todos si no se indica. */
FUNCTION getTiposViaProv (pNomCliPro in VARCHAR2, pCodPro in INTEGER)
    RETURN REF CURSOR;

/* Comprueba la dirección normalizada a nivel de población. */
PROCEDURE checkDirViaPob (pNomCliPro in VARCHAR2, pGetNivelCP in INTEGER,
    pTipoPob in INTEGER, pIdPob in INTEGER, pIdTVia in out INTEGER,
    pNomVia in out VARCHAR2, pCodPro in out INTEGER, pCP in out VARCHAR2,
    pPortal in out INTEGER, pCalif in out VARCHAR2, pBloque in out VARCHAR2,
    pEscale in out VARCHAR2, pPiso in out INTEGER, pPuerta in out VARCHAR2,
    pBuzon in out VARCHAR2, pNota in out VARCHAR2, pNivelDir out INTEGER,
    pNomNivel out VARCHAR2, pNomPro out VARCHAR2, pNomMuni out VARCHAR2,
    pNomECol out VARCHAR2, pNomESin out VARCHAR2, pNomNucleo out VARCHAR2,
    pTipoVia out VARCHAR2, pDistrito out INTEGER, pSeccion out INTEGER,
    pLetraSecc out VARCHAR2, pSubSecc out VARCHAR2, pIdNVia out INTEGER,
    pIdNPsVia out INTEGER, pNivelCP out INTEGER);

```

```

/* Crea si no existe y retorna el identificador de la dirección normalizada a nivel de población. */
FUNCTION getldDirViaPob (pNomCliPro in VARCHAR2, pGetNivelCP in INTEGER,
    pTipoPob in INTEGER, pIdPob in INTEGER, pIdTVia in out INTEGER,
    pNomVia in out VARCHAR2, pCodPro in out INTEGER, pCP in out VARCHAR2,
    pPortal in out INTEGER, pCalif in out VARCHAR2, pBloque in out VARCHAR2,
    pEscale in out VARCHAR2, pPiso in out INTEGER, pPuerta in out VARCHAR2,
    pBuzon in out VARCHAR2, pNota in out VARCHAR2, pNivelDir out INTEGER,
    pNomNivel out VARCHAR2, pNomPro out VARCHAR2, pNomMuni out VARCHAR2,
    pNomECol out VARCHAR2, pNomESin out VARCHAR2, pNomNucleo out VARCHAR2,
    pTipoVia out VARCHAR2, pDistrito out INTEGER, pSeccion out INTEGER,
    pLetraSecc out VARCHAR2, pSubSecc out VARCHAR2, pIdNVia out INTEGER,
    pIdNPsVia out INTEGER, pNivelCP out INTEGER) RETURN INTEGER;

```

```

/* Comprueba la dirección normalizada a nivel de provincia. */
PROCEDURE checkDirViaProv (pNomCliPro in VARCHAR2, pGetNivelCP in INTEGER,
    pNomPob in out VARCHAR2, pIdTVia in out INTEGER,
    pNomVia in out VARCHAR2, pCodPro in out INTEGER, pCP in out VARCHAR2,
    pPortal in out INTEGER, pCalif in out VARCHAR2, pBloque in out VARCHAR2,
    pEscale in out VARCHAR2, pPiso in out INTEGER, pPuerta in out VARCHAR2,
    pBuzon in out VARCHAR2, pNota in out VARCHAR2, pNivelDir out INTEGER,
    pNomNivel out VARCHAR2, pNomPro out VARCHAR2, pNomMuni out VARCHAR2,
    pNomECol out VARCHAR2, pNomESin out VARCHAR2, pNomNucleo out VARCHAR2,
    pNomTVia out VARCHAR2, pDistrito out INTEGER, pSeccion out INTEGER,
    pLetraSecc out VARCHAR2, pSubSecc out VARCHAR2, pTipoPob out INTEGER,
    pIdPob out INTEGER, pIdNVia out INTEGER, pIdNPsVia out INTEGER,
    pNivelCP out INTEGER);

```

```

/* Crea si no existe y retorna el identificador de la dirección normalizada a nivel de provincia. */
FUNCTION getldDirViaProv (pNomCliPro in VARCHAR2, pGetNivelCP in INTEGER,
    pNomPob in out VARCHAR2, pIdTVia in out INTEGER,
    pNomVia in out VARCHAR2, pCodPro in out INTEGER, pCP in out VARCHAR2,
    pPortal in out INTEGER, pCalif in out VARCHAR2, pBloque in out VARCHAR2,
    pEscale in out VARCHAR2, pPiso in out INTEGER, pPuerta in out VARCHAR2,
    pBuzon in out VARCHAR2, pNota in out VARCHAR2, pNivelDir out INTEGER,
    pNomNivel out VARCHAR2, pNomPro out VARCHAR2, pNomMuni out VARCHAR2,
    pNomECol out VARCHAR2, pNomESin out VARCHAR2, pNomNucleo out VARCHAR2,
    pNomTVia out VARCHAR2, pDistrito out INTEGER, pSeccion out INTEGER,
    pLetraSecc out VARCHAR2, pSubSecc out VARCHAR2, pTipoPob out INTEGER,
    pIdPob out INTEGER, pIdNVia out INTEGER, pIdNPsVia out INTEGER,
    pNivelCP out INTEGER) RETURN INTEGER;

```

```

/* Retorna los códigos INE de vía y de poblaciones, el código postal y el nivel de normalización del código postal de la dirección indicada. */
PROCEDURE getCodigosGeoPosDir (pNomCliPro in VARCHAR2, pldDir in INTEGER,
    pCP in VARCHAR2, pGetCodes in INTEGER, pGetNivelCP in INTEGER,
    pGetGeoPos in INTEGER, pNivelDir out INTEGER, pCodProv out INTEGER,
    pCodMuni out INTEGER, pCodEcol out INTEGER, pCodEsin out INTEGER,
    pCodNuc out INTEGER, pCodVia out INTEGER, pCPsVia out INTEGER,
    pNivelCP out INTEGER, pGeoLati out NUMBER, pGeoLong out NUMBER);

```

*/\* Retorna los datos de la dirección indicada. \*/*

```
PROCEDURE getDireccion (pNomCliPro in VARCHAR2, pGetNivelCP in INTEGER,  
    pIdDir in out INTEGER, pCodPro out INTEGER, pCP out VARCHAR2,  
    pPortal out INTEGER, pCalif out VARCHAR2, pBloque out VARCHAR2,  
    pEscale out VARCHAR2, pPiso out INTEGER, pPuerta out VARCHAR2,  
    pBuzon out VARCHAR2, pNota out VARCHAR2, pNivelDir out INTEGER,  
    pNomNivel out VARCHAR2, pNomPro out VARCHAR2, pNomMuni out VARCHAR2,  
    pNomECol out VARCHAR2, pNomESin out VARCHAR2, pNomNucleo out VARCHAR2,  
    pNomTVia out VARCHAR2, pNomVia out VARCHAR2, pDistrito out INTEGER,  
    pSeccion out INTEGER, pLetraSecc out VARCHAR2, pSubSecc out VARCHAR2,  
    pNivelCP out INTEGER);
```

*/\* Retorna el rango de direcciones que hay en el servidor. \*/*

```
PROCEDURE getRangoDirecciones (pNomCliPro in VARCHAR2, pMinDir out INTEGER,  
    pMaxDir out INTEGER);
```

*/\* Retorna el número de entidades singulares en las que está físicamente la vía o pseudovía indicada. \*/*

```
FUNCTION getNumESinsViaNuc (pNomCliPro in VARCHAR2,  
    pIdNVia in INTEGER, pIdNPsVia in INTEGER) RETURN INTEGER;
```

*/\* Retorna el número de entidades singulares en las que está físicamente la vía o pseudovía de la dirección indicada. \*/*

```
FUNCTION getNumESinsViaDir (pNomCliPro in VARCHAR2, pIdDir in INTEGER)  
    RETURN INTEGER;
```

*/\* Comprueba si el dominio Web existe y está activo en el servidor. \*/*

```
FUNCTION checkDominio (pNomCliPro in VARCHAR2, pDominio in VARCHAR2)  
    RETURN INTEGER;
```

*/\* Retorna la palabra de paso, las claves y el flag de seguridad, y la clave única o el identificador de cliente de la API de Google Maps para el dominio Web. \*/*

```
PROCEDURE getKeysDominio (pNomCliPro in VARCHAR2, pDominio in VARCHAR2,  
    pWord out VARCHAR2, pPar1 out INTEGER, pPar2 out INTEGER, pPar3 out INTEGER,  
    pPar4 out INTEGER, pFlag out INTEGER, pGMaps out VARCHAR2);
```

*\* Comprueba el nombre y la contraseña del usuario de la aplicación Web de administración. \*/*

```
FUNCTION checkWebUserPassword (pNomCliPro in VARCHAR2, pCurrUser in VARCHAR2,  
    pCurrPass in VARCHAR2) RETURN INTEGER;
```