***Entrega e Instalación en Test del Componente de Administración de Direcciones***

*v1.0*

13 Diciembre 2019

Para cualquier consulta o cuestión pendiente sobre la oferta la persona de contacto es:

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Teléfono** | **e-mail** |
| José Vicente Higón Valero | +34 600 55 44 74 | [jvhigon@gvsig.com](mailto:jvhigon@gvsig.com) |

Información general:

Asociación gvSIG

CIF G98125362

Calle Ángel Guimerá 61, puerta 3

[http://www.gvsig.com](http://www.gvsig.com/)

**Control de cambios**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Versión** | **Responsable** | **Organismo** | **Descripción del cambio** | **Fecha** |
| v.1.0 | Nacho Brodin | Asoc gvSIG | Documento inicial | 13/12/19 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Índice**

[**Introducción**](#_alce85kgbfwr) **4**

[**Descripción del componente**](#_117oeem8bn99) **4**

[**Descripción del API**](#_jdaczgho8g31) **6**

[Validación en los servicios del API](#_yujrx0cdm28a) 6

[Llamadas del API](#_lf79ds5x1q3d) 7

[**Despliegue del componente**](#_nrguegypqbpy) **12**

# Introducción

El siguiente documento sirve como entregable de instalación con la información del componente de administración de direcciones que no está contenida en el análisis y especificación de casos de uso. Básicamente se hará una descripción más detallada del componente así como del API y alguna información sobre el despliegue.

# Descripción del componente

El componente de administración de direcciones es un plugin de gvSIG Online que aporta la funcionalidad de envío de sugerencias de cambio sobre la base de datos de callejero tanto a nivel alfanumérico como geométrico. Este plugin es dependiente del de Geocodificación por lo que es obligatorio que este esté puesto.

El análisis y especificaciones de uso puede encontrarse en el documento 1.5.2-Analisis\_Y\_Especificacion\_De\_Casos\_De\_Uso\_Admin\_Direcciones\_v1.0 del primer entregable del proyecto.

Desde el punto de vista del usuario aporta una barra de herramientas que permite:

* **Reportar sugerencias de modificación de datos existentes**. Esta funcionalidad actúa sobre la última búsqueda realizada sobre callejero. Al accionarla se cargan los datos de la búsqueda en el panel de edición y sobre el mapa las geometrías de la búsqueda. Si es una calle se cargarán los tramos y la ubicación, si es un portal la ubicación y si es un municipio la geometría de este y la ubicación. La ubicación designa el punto donde aparece el popup cuando se hace una búsqueda en callejero.

La edición se realiza pinchando sobre las geometrías. La geometría cambia a color azul y permite su edición. Al acabar hay que volver a pinchar fuera del tramo para finalizar su edición. De esta forma volverá a su color naranja original.

Añadir tramos se realiza con el botón “Añadir Tramo”. Pinchando este se vuelve amarillo y permite la introducción de un tramo en el mapa. Al acabar vuelve a su color original. Si se quiere introducir otro tramo hay que volver a activarlo.

La ubicación se edita pinchando sobre el punto. Este vuelve azul y permite moverlo pinchando y arrastrando sobre él. Al igual que los tramos se finaliza la edición pinchando fuera de estos para volver a su color naranja original.

* **Envío de nuevos portales asociados a calles existentes**. Esta funcionalidad actúa sobre la última búsqueda realizada sobre callejero. Al accionarla se cargan los datos de la calle buscada en el panel de edición. Pinchando sobre el mapa directamente podemos situar la ubicación del portal y sobre el panel introducir el número de portal y código postal asociado.
* **Envío de nuevas calles**. Puede activarse la opción sin realizar una búsqueda previa. Un panel de edición de calle se abre pero sin datos sobre él. Será el usuario el encargado de rellenarlo.

Añadir tramos se realiza con el botón “Añadir Tramo”. Pinchando este se vuelve amarillo y permite la introducción de un tramo en el mapa. Al acabar vuelve a su color original. Si se quiere introducir otro tramo hay que volver a activarlo.

La ubicación de la calle se calculará automáticamente en servidor al realizar el envió y no es necesario introducirla.

Es obligatorio la introducción de un comentario de aceptación o rechazo antes de enviar la sugerencia de cambio.

Desde el punto de vista del validador aporta una entrada de menú “Gestión de Direcciones” con un submenú “Validación de direcciones” que permite listar las sugerencias pendientes de validación, aceptadas o rechazadas. La primera permite ver una previsualización en un mapa de todas las geometrías de las sugerencias pendientes pero sin interactuar con ellas. Al entrar en la validación de una sugerencia concreta el interfaz permitirá:

* **Comparar los datos alfanuméricos** **de la sugerencia** propuesta con los del registro original. El la parte izquierda y sin posibilidad de edición se muestra el registro original.
* **Modificar los datos alfanuméricos de la sugerencia**. En la parte derecha se muestran los datos de la sugerencia.
* **Restaurar los datos alfanuméricos** de algún campo del registro sobre la sugerencia. Junto a cada campo del registro original hay un botón que permite copiar el dato a la sugerencia.
* **Modificar las geometrías de los tramos de la sugerencia** sobre el mapa. La edición se realiza pinchando sobre ellos. La geometría cambia a color azul y permite su edición. Al acabar hay que volver a pinchar fuera del tramo para finalizar su edición. De esta forma volverá a su color naranja original.
* **Cambiar la ubicación del elemento sobre el mapa**. Los elementos (calles, portales, municipios) representan su ubicación por un punto. Pinchando sobre este vuelve azul y permite moverlo pinchando y arrastrando sobre él. Al igual que los tramos se finaliza la edición pinchando fuera de estos para volver a su color naranja original.
* **Rechazar la sugerencia de cambio** en la base de datos de Callejero. Esta opción hace que la sugerencia pase a rechazadas.
* **Aceptar la sugerencia de cambio** en la base de datos de Callejero. Esta opción hace que la sugerencia pase a aceptadas y los cambios se incorporen en la base de datos de callejero. Estos no tendrán un efecto inmediato en las búsquedas sino que está supeditado a la regeneración de las vistas materializadas sobre las que tira el motor de búsquedas.

# Descripción del API

Los servicios de API propuestos sobre la administración de direcciones propuestos en el documento de análisis del entregable 1 (1.5.2-Analisis\_Y\_Especificacion\_De\_Casos\_De\_Uso\_Admin\_Direcciones\_v1.0) son los siguientes:

* Reportar cambios
* Nueva dirección de portal
* Nuevo tramo de calle
* Nueva calle
* Obtener listado de sugerencias
* Aceptar sugerencia
* Rechazar sugerencia

La instalación de gvSIGOnline que lleva el plugin de administración de direcciones dispone de un swagger con la descripción de los servicios. Hay que destacar que para que esto sea así es necesario disponer del plugin restapi previamente instalado.

Todas las llamadas forman parte de la versión 1 del API por lo que llevarán la ruta {URL BASE}/api/v1/manageaddresses. En los siguientes apartados se documentan las llamadas del API y cómo acceder a ellas:

Los servicios del API funcionan con validación por token JWT y están disponibles para los usuarios con permisos. Es decir, si un usuario tiene permiso para acceder a una determinada funcionalidad desde gvSIG Online también lo tiene desde el API y viceversa.

## Validación en los servicios del API

Para la obtención del token hay que hacer usar la siguiente llamada

/auth/api-token-auth/

suministrando en el body los datos de validación con el siguiente formato

|  |
| --- |
| {  "username": "string",  "password": "string"  } |

Esto devolverá un token con el aspecto siguiente:

|  |
| --- |
| {  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6Im5icm9kaW4iLCJvcmlnX2lhdCI6MTU3NjIyMjA4NiwidXNlcl9pZCI6MTQsImVtYWlsIjoibmJyb2RpbkBzY29sYWIuZXMiLCJleHAiOjE1NzYzOTQ4ODZ9.dWSl37JAvcfKlOmIVXEJOJw40EK5kFaa66bWyjbzs6s"  } |

Este token se utilizará en cada llamada al API insertandolo en la cabecera con el parámetro authorization. Un ejemplo de como quedaría una llamada al servicio con Curl es el siguiente:

|  |
| --- |
| curl -X GET "https://localhost/gvsigonline/api/v1/manageaddresses/suggestions/" -H "accept: application/json" -H "Authorization: JWT eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6Im5icm9kaW4iLCJvcmlnX2lhdCI6MTU3NjIyMjA4NiwidXNlcl9pZCI6MTQsImVtYWlsIjoibmJyb2RpbkBzY29sYWIuZXMiLCJleHAiOjE1NzYzOTQ4ODZ9.dWSl37JAvcfKlOmIVXEJOJw40EK5kFaa66bWyjbzs6s" |

Si se quiere hacer uso del API a través del Swagger habrá que introducir la cadena

JWT eyJhb...resto del token...yjbzs6s dentro de la caja de diálogo que aparece al pulsar el botón Authorize. A partir de ese momento todas las llamadas que se realicen usarán la validación con el token introducido.

## Llamadas del API

* Obtener listado de sugerencias

Es necesario tener permiso de validación para el uso de esta llamada. Para la obtención del listado de sugerencias pendientes de validación hay que usar el servicio sin parámetros

/api/v1/manageaddresses/suggestions/

Este devolverá un JSON con la lista de sugerencias. Cada una contiene los siguientes campos:

{

|  |  |
| --- | --- |
| id | integer  *title: ID*  *readOnly: true* |
| idcalle | string  *title: Idcalle*  *maxLength: 250*  *minLength: 1*  *x-nullable: true* |
| nombre\* | string  *title: Nombre*  *minLength: 1* |
| localidad | string  *title: Localidad*  *maxLength: 1024*  *minLength: 1* |
| departamento | string  *title: Departamento*  *maxLength: 250*  *minLength: 1* |
| lat | number  *title: Lat*  *x-nullable: true* |
| lon | number  *title: Lon*  *x-nullable: true* |
| numero | integer  *title: Numero*  *maximum: 2147483647*  *minimum: -2147483648*  *x-nullable: true* |
| cp | integer  *title: Cp*  *maximum: 2147483647*  *minimum: -2147483648* |
| comentario | string  *title: Comentario*  *x-nullable: true* |
| fechaalta | string($date-time)  *title: Fechaalta*  *readOnly: true* |
| reason | string  *title: Reason*  *minLength: 1*  *x-nullable: true* |
| status | integer  *title: Status* |
| type | integer  *title: Type* |
|  |  |

}

* Lista de tramos asociados a una sugerencia

Esta llamada complementa el servicio anterior aportando las geometrías asociadas a cada sugerencia. Para ello se usa la llamada:

/api/v1/manageaddresses/suggestions/{sug\_id}/tramos/

donde {sug\_id} es el identificador de la sugerencia de la cual se quieren obtener los tramos.

El objeto devuelto es un JSON con la lista de tramos de esta forma:

Stretch{

|  |  |
| --- | --- |
| id | integer  *title: ID*  *readOnly: true* |
| geometry\* | string  *title: Geometry* |
| issuggestion | boolean  *title: Issuggestion*  Verdadero si el tramo es de la sugerencia y falso si el tramo forma parte de la geometría original |
| idchange\* | integer  *title: Idchange* |
|  |  |

}

* Reportar cambios, nueva dirección de portal, nuevo tramo de calle, nueva calle

Las cuatro funciones quedan cubiertas con una llamada al API usando distintos parámetros:

/api/v1/manageaddresses/suggestion/

Como parámetro se introduce un JSON cuyos campos se explican a continuación

NewSuggestion{

|  |  |
| --- | --- |
| type | integer  *title: Type*  *default: 0*  0: modificación, 1: nueva dirección, 2: nueva calle |
| idcalle\* | string  *title: Idcalle*  *maxLength: 250*  *minLength: 1*  Identificador de la calle si es nuevo portal o nuevo tramo de calle |
| snombre\* | string  *title: Snombre*  *maxLength: 250*  *minLength: 1*  Nombre de la calle |
| slat\* | number  *title: Slat*  Latitud del elemento |
| slon\* | number  *title: Slon*  Longitud del elemento |
| snumero\* | integer  *title: Snumero*  Número del portal |
| scp\* | integer  *title: Scp*  Código postal del portal |
| smunicipio\* | string  *title: Smunicipio*  *maxLength: 1024*  *minLength: 1*  Municipio donde se encuentra la calle |
| sdepartamento\* | string  *title: Sdepartamento*  *maxLength: 250*  *minLength: 1*  Departamento donde se encuentra la calle |
| stramos | [  Lista de tramos de la calle. Cada tramo es una cadena en formato WKT  string  *minLength: 1*] |
| comentario | string  *title: Comentario*  *default: Insertado desde el API*  *maxLength: 1024*  *minLength: 1*  Comentario del usuario que lo dio de alta |
| onombre | string  *title: Onombre*  *maxLength: 250*  *minLength: 1*  Nombre original de la calle  Default: snombre |
| olat | number  *title: Olat*  Latitud original del elemento  Default: slat |
| olon | number  *title: Olon*  Longitud original del elemento  Default: slon |
| onumero | integer  *title: Onumero*  Número original del portal  Default: snumero |
| ocp | integer  *title: Ocp*  Código postal original del portal  Default: scp |
| omunicipio | string  *title: Omunicipio*  *maxLength: 1024*  *minLength: 1*  Municipio original donde se encuentra la calle  Default: smunicipio |
| odepartamento | string  *title: Odepartamento*  *maxLength: 250*  *minLength: 1*  Departamento original donde se encuentra la calle  Default: sdepartamento |
| otramos | [  Lista de tramos originales de la calle. Cada tramo es una cadena en formato WKT  string  *minLength: 1*] |
|  |  |

}

Los campos con asterisco en rojo son obligatorios para cualquier tipo de sugerencia. El parámetro type define si es nueva calle o dirección o una modificación sobre los datos del callejero existente.

Los campos que comienzan por “s” son los que corresponden a los datos de sugerencia, es decir a los que ha propuesto el usuario, y los que empiezan por “o” los del registro original. Por ejemplo, si es nueva calle el campo “snombre” llevará el nombre de la nueva calle propuesta y el campo “onombre” irá vacío debido a que la calle no existía por lo que no hay registro original. Sin embargo, si se trata de una propuesta de modificación del nombre de una calle, el campo “snombre” llevará el nombre de la propuesta de cambio y el “onombre” llevará el nombre original de la calle.

En los casos que son modificaciones de registros existentes no siempre es obligatorio introducir los datos originales si estos no se modifican. Es decir, si el campo “snombre” no ha cambiado el sistema supone que “onombre” contiene la misma información.

* Aceptar sugerencia

Para la aceptación de una sugerencia se usa el servicio

/api/v1/manageaddresses/acceptsuggestion/{sug\_id}/

donde {sug\_id} es el identificador de la sugerencia a aceptar

* Rechazar sugerencia

Para la aceptación de una sugerencia se usa el servicio

/api/v1/manageaddresses/acceptsuggestion/{sug\_id}/

donde {sug\_id} es el identificador de la sugerencia a rechazar

# Despliegue del componente

El despliegue en test del componente se realiza sobre gvSIG online existente accesible desde <https://visor-test.ide.uy/ideuy/> como una extensión de este. Por ello, para la instalación de este plugin se hará una actualización del core de gvSIG y se añadirán dos nuevas extensiones, plugin\_resapi y plugin\_manageaddresses. Además es necesario que esté instalado el plugin de geocoding del cual depende la gestión de direcciones. Este ya existía en el visor por lo que no se instalará.

La actualización de gvSIG online se realiza mediante el script de instalación estándar. En el proyecto de despliegue app\_ideuy habrá que añadir los nuevos plugins en el fichero environment.test a la variable GVSIGOL\_PLUGINS. Se añadirá 'gvsigol\_plugin\_restapi' y 'gvsigol\_plugin\_manageaddresses' a la lista. Dentro del mismo fichero se añadirán las variables de entorno que contienen los datos de conexión a la base de datos

DBHOST\_DIRECCIONES="10.255.3.224"

DBPORT\_DIRECCIONES=5432

DBNAME\_DIRECCIONES="tsubasa"

DBSCHEMA\_DIRECCIONES="tsubasa2"

DBUSER\_DIRECCIONES="geomatica"

DBPASSWORD\_DIRECCIONES="\*\*\*\*\*\*\*"

El script de despliegue sustituirá el valor de estas variables en el settings del plugin de gvsig en el momento del despliegue. De esta forma no se guardan en los fuentes en el repositorio datos de conexión ni passwords específicos de un despliegue.

Un fichero configure.sh dentro de la propia extensión es utilizado por el instalador para la sustitución de las variables. Este es un shell script de la siguiente forma:

|  |
| --- |
| #!/bin/bash  function configure() {  grep -rl "##DBHOST\_DIRECCIONES##" | xargs sed -i "s ##DBHOST\_DIRECCIONES## $DBHOST\_DIRECCIONES g"  grep -rl "##DBPORT\_DIRECCIONES##" | xargs sed -i "s ##DBPORT\_DIRECCIONES## $DBPORT\_DIRECCIONES g"  grep -rl "##DBNAME\_DIRECCIONES##" | xargs sed -i "s ##DBNAME\_DIRECCIONES## $DBNAME\_DIRECCIONES g"  grep -rl "##DBSCHEMA\_DIRECCIONES##" | xargs sed -i "s ##DBSCHEMA\_DIRECCIONES## $DBSCHEMA\_DIRECCIONES g"  grep -rl "##DBUSER\_DIRECCIONES##" | xargs sed -i "s ##DBUSER\_DIRECCIONES## $DBUSER\_DIRECCIONES g"  grep -rl "##DBPASSWORD\_DIRECCIONES##" | xargs sed -i "s ##DBPASSWORD\_DIRECCIONES## $DBPASSWORD\_DIRECCIONES g"  }  function move\_template() {  mv settings\_tpl.py settings.py  }  configure  move\_template |