

# WSRepsal v1.0.21 - Guía de implementación

---

Web service público para el Registro Público de Empleadores con sanciones laborales

## Objetivo

El siguiente proyecto tiene por objetivo proveer un servicio web REST de acceso público para que la ciudadanía pueda consultar de manera programática la información que hoy publicamos a través del *Registro Público de Empleadores con Sanciones Laborales (REPSAL)*

## Prerequisitos

Tener instalada la última versión estable de git ([2.23.0](#)) y la última versión LTS de node ([v10.16.0](#))

## Implementación (Documento técnico de implementación)

Generar la aplicación a partir del repositorio de código fuente:

```
$ git clone --branch vX.Y.Z https://tfsqa.trabajo.gob.ar/tfs/sgte-  
it/trabajo_fiscalizacion_repsal/_git/wsrepsal wsrepsal --depth 1  
  
Cloning into 'wsrepsal'...  
[...]  
  
$ cd wsrepsal\src\  
  
$ npm run build
```

El proceso generará los siguientes archivos en la carpeta wsrepsal/dist:

- wsrepsal\_vX.Y.Z.exe: ejecutable del servicio
- config/: carpeta de configuración
  - default.example.json: archivo de configuración de ejemplo
- logs/: carpeta de logs
  - readme\_log.md

## Recursos expuestos por el servicio

El servicio expone los siguientes recursos:

- Web service REST en <http://wsrepsal.trabajo.gob.ar/publico/api/sanciones>
- Documentación del web service REST en <http://wsrepsal.trabajo.gob.ar/publico/api/docs>
- Recursos estáticos (html, css, js) en <http://wsrepsal.trabajo.gob.ar/publico>

## Recursos consumidos por el servicio

- Conexión a base de datos [entorno].LogsApp
  - configurado en: `./config/default.json` - `db_log_connection_string`
  - Requiere únicamente permiso de ejecución sobre el sp `WriteLog4Net`
- Conexión a base de datos [entorno].RepsalPub
  - configurado en: `./config/default.json` - `repsal_connection_string`
  - Utiliza la cuenta `wsRepsalPublico`, perteneciente al rol `rolwsRepsalPublico`
  - Requiere únicamente acceso de lectura a `RepsalPub.SancionesPublicadas`
- Proceso offline que alimenta la tabla `RepsalPub.SancionesPublicadas` (ver el apartado *Implementación de los cambios en la base de datos*)

## Componentes de software que usa el servicio

Se trata de un archivo ejecutable autónomo.

### Configuraciones a nivel servidor

Configurar la aplicación como un servicio de windows (ver el apartado *Instalación desde cero*)

### Configuraciones a nivel sitio

Las necesarias para exponer un servicio web de acceso público (configurar IIS como reverse proxy)

### Configuraciones a nivel servicio

Ver el apartado *Opciones de configuración de WSRepsal*

## Instalación desde cero

### Implementación del servicio web

1. Crear el directorio de la aplicación:

```
mkdir c:\w3-sites\wsrepsal
```

2. Generar el exe

```
$ git clone --branch vX.Y.Z https://tfsqa.trabajo.gob.ar/tfs/sgte-  
it/trabajo_fiscalizacion_repsal/_git/wsrepsal wsrepsal --depth 1  
  
Cloning into 'wsrepsal'...  
[...]
```

```
$ cd wsrepsal\src\  
  
$ npm run build
```

3. Configurar el archivo `config\default.json`
4. Crear e iniciar el servicio (con permisos de administrador)

```
nssm install wsrepsal c:\w3-sites\wsrepsal\wsrepsal.exe  
nssm start wsrepsal
```

Para especificar otro entorno de ejecución (desa. test, qa, prod) configurar la opción env en `config\default.json`, o mediante una variable de entorno `ENV` o pasar el parámetro por línea de comandos, ej:

```
c:\w3-sites\wsrepsal\wsrepsal --env=qa
```

## Implementación de los cambios en la base de datos

Ejecutar el script de sql `wsrepsal_implementacion_script.sql` en la base *RepsalPub*

El script efectúa las siguientes tareas:

1. Crea al rol *rolwsRepsalPublico*
2. Crea el login *wsRepsalPublico* si no existe  
Atención! Modificar el script para asignar un password seguro
3. Crea el usuario *wsRepsalPublico*
4. Crea la tabla *SancionesPublicadas*
5. Asigna permisos de lectura sobre la tabla *SancionesPublicadas* al rol *rolwsRepsalPublico*
6. Crea el stored procedure *LoadSancionesPublicadas*
7. Ejecuta el stored procedure *LoadSancionesPublicadas* para cargar inicialmente los datos en la tabla *SancionesPublicadas*

A fin de mantener actualizada la tabla *SancionesPublicadas*, es preciso agregar un step que ejecute el stored procedure `RepsalPub.dbo.LoadSancionesPublicadas` en el job *Repsal – Procesar Sanciones (Critico)* luego de `Repsal.dbo.ProcesoActualizar` y antes de volver a setear la base *RepsalPub* en modo read-only.

## Opciones de configuración de WSRepsal

Los archivos de configuración se encuentran en la carpeta `./config`.

Estos son los parámetros que pueden configurarse a través del archivo de configuración:

Parámetro	Valor por defecto	Descripción
env	desa	El entorno de la aplicación. Valores posibles: desa, test, qa, prod
repsal_connection_string		String de conexión a la base de datos de Repsal (RepsalPub)
ip	127.0.0.1	La dirección ip en la cual se podrá acceder al proxy
port	8000	El puerto a través del cual se podrá acceder al servicio web
api_url_prefix	/api	El prefijo a agregar al endpoint. Ej: /api/sanciones

Mediante los siguientes parámetros se pueden configurar las opciones para loguear eventos

Parámetro	Valor por defecto	Descripción
log_level	info	Nivel de logueo general, valores posibles: debug, verbose, info, warning, error
log_label	wsrepsal	Etiqueta por defecto para logs
console_log_level	verbose	Nivel de logueo para consola, valores posibles: debug, verbose, info, warning, error
file_log_level	info	Nivel de logueo para archivo, valores posibles: none, debug, verbose, info, warning, error
file_log	./logs/wsrepsal.log	El archivo al cual guardar los logs
db_log_level	info	Nivel de logueo para base de datos, valores posibles: none, debug, verbose, info, warning, error
db_log_connection_string	N/A	El string de conexión de la base de datos
db_log_sp	WriteLog4Net	El stored procedure a ejecutar para grabar los logs

## String de conexión

El usuario a utilizar es wsRepsalPublico, el cual pertenece al grupo rolwsRepsalPublico.

El único permiso que precisa el usuario wsRepsalPublico es permisos de lectura sobre la tabla RepsalPub.dbo.SancionesPublicadas.

El string de conexión soporta los siguientes formatos:

#### Estilo clásico

```
Server=localhost,1433;Database=database;User
Id=username;Password=password;Encrypt=true
Driver=msnodesqlv8;Server=
(local)\INSTANCE;Database=database;UID=DOMAIN\username;PWD=password;Encrypt=true
```

#### Estilo URI

```
mssql://username:password@localhost:1433/database?encrypt=true
mssql://username:password@localhost/INSTANCE/database?
encrypt=true&domain=DOMAIN&driver=msnodesqlv8
```

Nota:

Dado que los archivos de configuración son archivos .json, las barras correspondientes a los nombres de instancia `\` deben escaparse utilizando doble barra `\\`, al igual que en JavaScript. Ej:

```
{
  "repsal_connection_string": "Server=s1-dixx-
sql10\\instancia;Database=RepsalPub;Uid=wsRepsalPublico;Pwd=****",
  "port": 8080,
  [...]
```

Ejemplo de configuración (./config/default.example.json)

```
{
  "env": "desa",
  "repsal_connection_string": "Server=s1-dixx-
sql10\\a01;Database=RepsalPub;Uid=wsRepsalPublico;Pwd=****",
  "port": 8080,
  "api_url_prefix": "/api",
  "ip": "127.0.0.1",
  "log_level": "info",
  "log_label": "wsrepsal",
  "console_log_level": "debug",
  "file_log_level": "debug",
  "file_log": "./logs/wsrepsal.log",
  "db_log_level": "debug",
  "db_log_connection_string": "Server=S1-DIXX-
SQL10;Database=LogsApp;Uid=usrLogsApp;Pwd=****",
  "db_log_sp": "WriteLog4Net"
}
```

Consultando la versión del aplicativo

Si se ejecuta el servicio pasándole el parámetro `-v` o `--version` retornará la versión actual del ejecutable, y no iniciará el servicio.

```
C:\apps\wsrepsal>wsrepsal.exe --version
wsrepsal v1.0.21
node v10.15.3
os Windows_NT 6.1.7601 (x64)
```

## Especificando la configuración por línea de comandos

Los parámetros pueden ser especificados mediante línea de comandos, desde una variable de entorno del sistema o a través de los archivos de configuración, y tienen preponderancia en ese orden.

Ejemplo especificando una variable por línea de comando:

```
C:\apps\wsrepsal>wsrepsal.exe --env=qa --port=9000

INFO - reading configuration from .\config\default.json

WARN - configuration file .\config\qa.json not found

INFO - environment: qa

2019-08-14 20:34:40.40 [wsrepsal_app.ts_init] info: loading config for env qa from
.\config\default.json
  meta: {"app":"wsrepsal","env":"qa","configuration_files":
[".\config\default.json"]}

2019-08-14 20:34:40.40 [wsrepsal_app.ts_init] info: using configuration
  meta: {"app":"wsrepsal","env":"qa","repsal_connection_string":"Server=s1-dixx-
sql10;Database=RepsalPub_20190809;Uid=wsRepsalPublico;Pwd=*****","ip":"127.0.0.
1","port":9000,"api_url_prefix":"/api","log_level":"info","log_label":"wsrepsal","
console_log_level":"debug","file_log_level":"debug","file_log":"./logs/wsrepsal.lo
g","db_log_level":"debug","db_log_connection_string":"Server=S1-DIXX-
SQL10;Database=LogsApp;Uid=usrLogsApp;Pwd=*****","db_log_sp":"WriteLog4Net"}

2019-08-14 20:34:40.40 [wsrepsal_routes.ts_routes] info: attaching
wsrepsal.sanciones endpoint to /api/sanciones
  meta: {"app":"wsrepsal","env":"qa","endpoint":"/api/sanciones"}

2019-08-14 20:34:40.40 [wsrepsal_app.ts_init] info: wsrepsal listening on
127.0.0.1:9000/api
  meta: {"app":"wsrepsal","env":"qa"}
```

## Especificando la configuración por variables de entorno

Lo mismo podemos lograr utilizando variables de entorno de sistema. La variable debe llamarse igual que el parámetro a configurar.

```
C:\apps\wsrepsal>set env=qa
C:\apps\wsrepsal>set port=9000
C:\apps\wsrepsal>wsrepsal.exe

[...]
```

```
2019-08-14 20:34:40.40 [wsrepsal_app.ts_init] info: using configuration
  meta: {"app":"wsrepsal","env":"qa","repsal_connection_string":"Server=s1-dixx-
sql10;Database=RepsalPub_20190809;Uid=wsRepsalPublico;Pwd=*****","ip":"127.0.0.
1","port":9000,"api_url_prefix":"/api","log_level":"info","log_label":"wsrepsal","
console_log_level":"debug","file_log_level":"debug","file_log":"./logs/wsrepsal.lo
g","db_log_level":"debug","db_log_connection_string":"Server=S1-DIXX-
SQL10;Database=LogsApp;Uid=usrLogsApp;Pwd=*****","db_log_sp":"WriteLog4Net"}

[...]
```

```
2019-08-14 20:34:40.40 [wsrepsal_app.ts_init] info: wsrepsal listening on
127.0.0.1:9000/api
  meta: {"app":"wsrepsal","env":"qa"}
```

## Rutina de mantenimiento de paquetes

---

Ejecutar el siguiente comando para escanear las dependencias del proyecto en busca de vulnerabilidades:

```
$ git clone --branch vX.Y.Z https://tfsqa.trabajo.gob.ar/tfs/sgte-
it/trabajo_fiscalizacion_repsal/_git/wsrepsal wsrepsal --depth 1

$ cd wsrepsal\src\

$ npm audit

=== npm audit security report ===

found 0 vulnerabilities
in 883242 scanned packages
```

En caso de hallar vulnerabilidades, se puede intentar resolverlas ejecutando:

```
$ npm audit fix
```

Si luego de ejecutar el comando `npm audit fix` sigue habiendo vulnerabilidades, es preciso efectuar un análisis específico de la situación para decidir los pasos a tomar, a saber: eliminar la dependencia, reemplazarla por otra librería, informar al desarrollador del paquete o ignorar la advertencia.

## Probar la aplicación

---

### Ejecución de las pruebas unitarias

```
$ git clone --branch vX.Y.Z https://tfsqa.trabajo.gob.ar/tfs/sgte-
it/trabajo_fiscalizacion_repsal/_git/wsrepsal wsrepsal --depth 1

Cloning into 'wsrepsal'...
[...]

$ cd wsrepsal\src\

$ npm install

$ npm run test
```

### Prueba de la api rest

```
λ curl -i -X GET "http://localhost:8080/api/sanciones?
page=1&len=2&fields=sancion_id,cuit,razon_social,actividad&order=cuit"
HTTP/1.1 200 OK
X-Powered-By: Express
Access-Control-Allow-Origin: *
Access-Control-Allow-Headers: Origin, X-Requested-With, Content-Type, Accept
Access-Control-Expose-Headers: X-Total-Count, My-Header
Content-Type: application/json; charset=utf-8
x-total-count: 46330
Content-Length: 508
ETag: W/"1fc-E6QRalrLleEXjJNJUWgwk0qx7Cs"
Date: Wed, 14 Aug 2019 23:38:01 GMT
Connection: keep-alive

[
  {
    "sancion_id": "4DF1B0CA-D2F2-4C5C-BF5D-C7291F291DD6",
    "cuit": "20005757844",
    "razon_social": "PUPPO HECTOR JORGE",
    "actividad": "Comercio al por mayor y menor, actividades inmobiliarias,
empresariales y de alquiler, actividades d"
  },
  {
    "sancion_id": "33E6D91D-717F-4F32-A7AA-BF87E1362792",
    "cuit": "20019921396",
```



```
"razon_social": "ALES ANTONIO",  
"actividad": "Comercio al por mayor y menor, actividades inmobiliarias,  
empresariales y de alquiler, actividades d"  
}  
]
```

## Prueba de la documentación de la api rest

Navegar a <http://localhost:8080/api/docs>

## Prueba de la ui de consulta

Navegar a <http://localhost:8080/>

Instalación de curl utilizando el gestor de paquetes [Chocolatey](#)

1. Instalar Chocolatey ([source](#))
2. Instalar curl

```
$ choco install curl
```

## Instalar wsrepsal como un servicio de windows

---

1. Instalar NSSM - Non-Sucking Service Manager

```
choco install nssm
```

o descargarlo desde: <https://nssm.cc/release/nssm-2.24.zip>

2. Instalar el servicio

Para instalar el servicio mediante la interfaz gráfica, simplemente ejecutar:

```
nssm install
```

Para configurar el servicio por línea de comandos:

Suponiendo que el servicio está en [c:\w3-sites\wsrepsal\wsrepsal.exe](#)

Ejecutar los siguientes comandos como administrador

```
nssm install wsrepsal c:\w3-sites\wsrepsal\wsrepsal.exe
```

```
nssm set wsrepsal Application c:\w3-sites\wsrepsal\wsrepsal.exe

nssm set wsrepsal AppDirectory c:\w3-sites\wsrepsal

nssm set wsrepsal DisplayName "wsrepsal"

nssm set wsrepsal Description "Servicio web REST del Registro Público de Empleadores con Sanciones Laborales (REPSAL)"

nssm set wsrepsal Start SERVICE_AUTO_START

nssm set wsrepsal AppThrottle 1500

nssm set wsrepsal AppExit Default Restart

nssm set wsrepsal AppRestartDelay 0
```

**Nota:** se recomienda ejecutar el servicio con una cuenta que no tenga permisos de admin:

```
nssm set wsrepsal ObjectName trabajo\appwsrepsal pwd
nssm set wsrepsal Type SERVICE_WIN32_OWN_PROCESS
```

Para ejecutar con la cuenta local del sistema:

```
nssm set wsrepsal ObjectName LocalSystem
nssm set wsrepsal Type SERVICE_WIN32_OWN_PROCESS
```

Para más opciones consultar [la documentación](#) en línea.

### 3. Eliminar el servicio

```
nssm remove wsrepsal
```

### 4. Iniciar, detener, reiniciar servicio

```
nssm start wsrepsal

nssm stop wsrepsal

nssm restart wsrepsal
```

**Nota:** el puerto en el que escucha el servicio wsrepsal debe ser accesible sólo por el servicio de IIS.