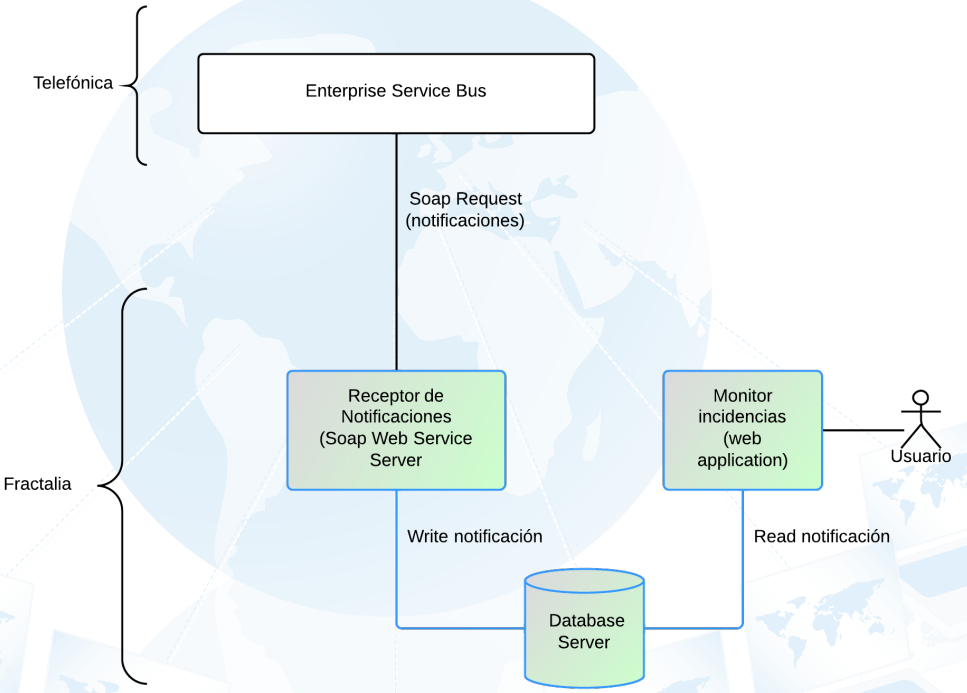
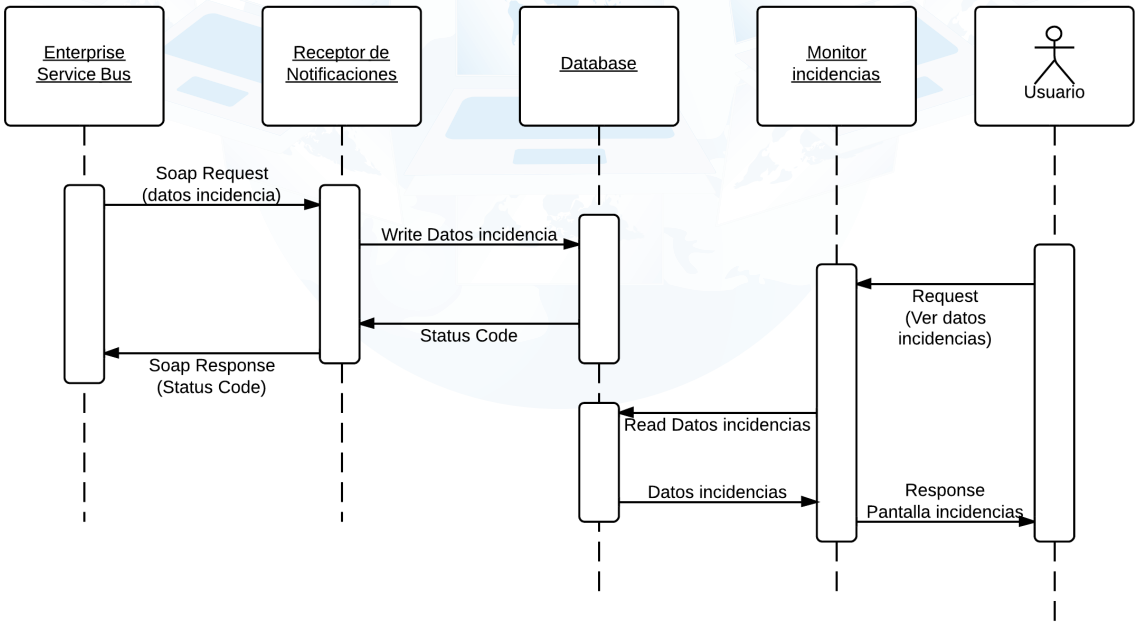
Manual de usuario SGSD

Despliegue de la aplicación





FRACTALIA

Instalación de la aplicación

Llamaremos installDir al directorio donde se va a desplegar la aplicación.

Se trata de una aplicación desarrollada con Symfony2, así que se puede utilizar todo lo que este framework permite de cara al despliegue y configuración. No obstante, ofrecemos los siguientes pasos para facilitar la tarea.

Nota

Es indispensable que el usuario con el que se ejecuta el web server tenga permisos de escritura sobre los directorios app/cache y app/logs. Además, si se va a hacer uso de la consola de Symfony2 (lo cual será necesario para realizar tareas de mantenimiento) también debe tener permisos de escritura sobre dichos el usuario con el que se lancen los comandos de consola. Una forma de conseguir esta simultaneidad en los permisos de escritura es a través de la aplicación setfacl (http://symfony.com/doc/current/book/installation.html).

```
APACHEUSER=`ps aux | grep -E '[a]pache|[h]ttpd|[_]www|[w]ww-data' | grep -v root | head -1 | cut -d\  -f1`
sudo setfacl -R -m u:"$APACHEUSER":rwX -m u:`whoami`:rwX app/cache app/logs
sudo setfacl -dR -m u:"$APACHEUSER":rwX -m u:`whoami`:rwX app/cache app/logs
```

Pasos de instalación

- Desplegar los archivos de la aplicación:
- Dar permisos de escritura al usuario del web server en los archivos app/cache y app/logs.
- Editar el fichero de configuración app/config/parameters.yml y definir los valores de database_host, database_port, database_name, database_user, database_password con los valores para la conexión a la base de datos. El parámetro secret debe ser una cadena de caracteres cualquiera y sgsd_soap_cache sirve para especificar si se desea cachear los resultados de las peticiones SOAP (0 no se cachea, 1 si se cachea).
- Crear la base de datos a partir del volcado doc/basedatos.sql
- Crear un virtual host cuyo document root sea installDir/web. En esta web:

http://symfony.es/documentacion/como-configurar-bien-apache-para-las-aplicaciones-symfony2/, se explica una buena forma de hacer esto. No obstante ofrecemos aquí una posible configuración:

```
<VirtualHost *:80>
        ServerName      mi-sitio.com
        DocumentRoot    "installDir/web"
        DirectoryIndex  app.php

        <Directory "installDir/web">
                AllowOverride None
                Allow from All

                <IfModule mod_rewrite.c>
                        Options -MultiViews
                        RewriteEngine On
                        RewriteCond %{REQUEST_FILENAME} !-f
                        RewriteRule ^(.*)$ app.php [QSA,L]
                </IfModule>
        </Directory>

        KeepAlive           On
        MaxKeepAliveRequests 200
        KeepAliveTimeout     5

</VirtualHost>
```

Funcionamiento del servicio web de recepción de tickets

Se trata de un servicio web de tipo SOAP que implementa el servicio especificado en el archivo de descripción GINotificarEvento.wsdl, facilitado por el cliente para el desarrollo del proyecto.

El servicio consiste en una sola operación (notificarEvento) cuyos argumentos son los datos que constituyen una incidencia.

Se puede acceder al WSDL del servicio a través de la URL:

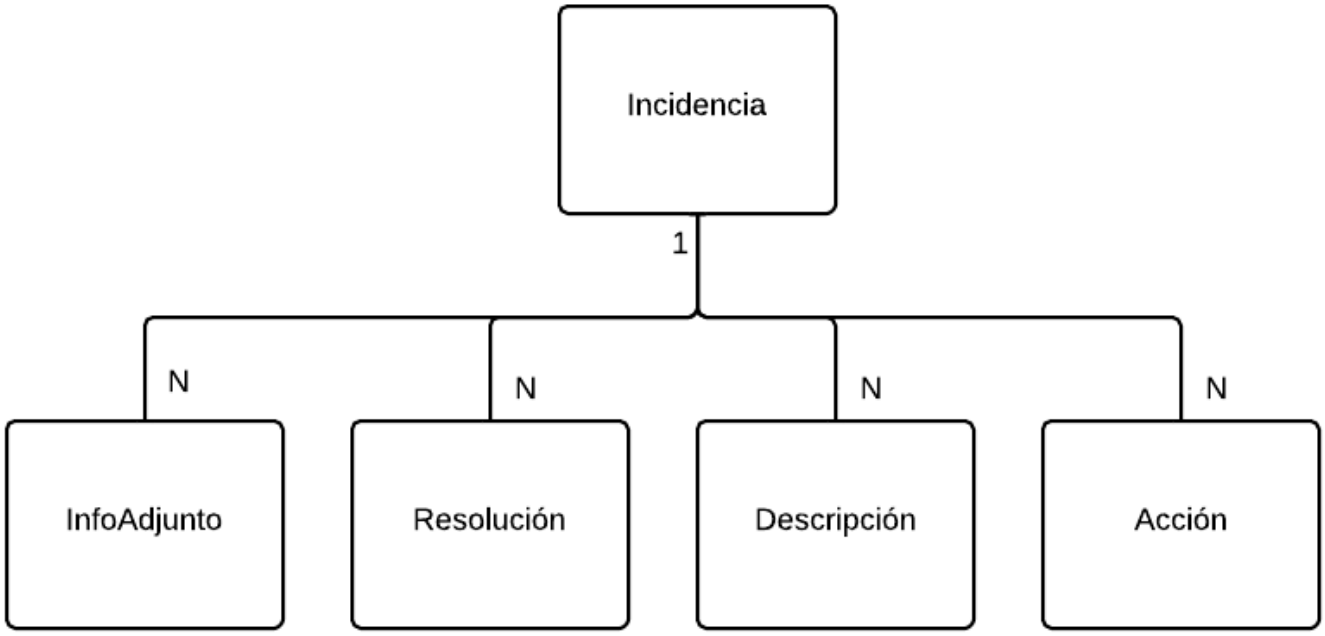
http://mi-sitio.com/wsdl

La respuesta ofrece todo los datos necesarios para construir una petición correcta al servicio.

La URL del servicio (que se especifica en la URL anterior) es:

http://mi-sitio.com/GINotificarEvento

Es ahí donde el cliente debe enviar la petición SOAP con los datos de la incidencia. El servicio convierte dichos datos en un objeto interno de la aplicación y sus agregados según se muestra en el siguiente modelo de datos:



El código de las clases que implementan este modelo se encuentra en src/Pi2/Fractalia/Entity/SGSD. Estas clases se encuentran mapeadas sobre la base de datos a través de un ORM (Doctrine2). La información de mapeo se puede ver en las anotaciones de las propias clases.

La lógica que implementa este proceso de convertir los datos de la petición SOAP en una serie de objetos (entidades) y persistir estas sobre la base de datos se encuentra fundamentalmente en los archivos:

- src/Pi2/Fractalia/SGSDSoapServerBundle/Controller/DefaultController.php
- src/Pi2/Fractalia/SGSDSoapServerBundle/Soap/SOAPAPI.php

Nota

A petición del cliente, las respuestas erróneas que emite el servicio no sigue el formato establecido en el estándar SOAP. En su lugar se devuelve lo siguiente:

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope" xmlns:ns1="http://service.gestionincidencias.telefonica.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <env:Body>
    <ns1:notificarEventoResponse>
        <ns1:returnCode>SOAP-ENV:Client</ns1:returnCode>
        <ns1:message>Bad Request</ns1:message>
        <ns1:IDCasoExterno xsi:nil="true"/>
        </ns1:notificarEventoResponse>
    </env:Body>
</env:Envelope>
```

Acelerador APC php

El APC, o caché alternativo de PHP (por sus siglas en inglés de Alternative PHP Cache), es un código de operación de caché libre y abierto para PHP. Su objetivo es el de proporcionar un marco robusto, libre y abierto para optimizar código de PHP intermedio mediante el almacenamiento en caché.

Su configuración es la siguiente:

en /etc/php5/apache2/conf.d/apc.ini

```
extension=apc.so
apc.shm_size=128M
```

Funcionamiento del monitor de incidencias

Las incidencias enviadas por el cliente SOAP y procesadas por el servicio anterior, se pueden consultar a través de un frontend de monitorización al que se accede desde la siguiente URL:

http://mi-sitio.com/monitor

Desde ahí se puede seleccionar el servicio cuyas incidencias se desean visualizar. El monitor se actualiza cada minuto realizando automáticamente una petición a la URL de la categoría seleccionada.

Configuración del monitor de incidencias

La definición de los servicios y los aspectos relacionados con los mismos (campos a mostrar, condiciones, intervalos horarios, etcétera) se configura a través del archivo app/config/web_monitor.yml.

Nota importante

Cualquier cambio que se realice sobre este fichero (web_monitor.yml) sólo tendrá efecto en la aplicación una vez borrada la caché de esta. El borrado de la cache se realiza de alguna de las siguientes formas:

```
installDir/app/console cache:clear

rm -rf installDir/app/cache/*
```

El fichero de configuración se llama: web_monitor.yml y se encuentra ubicado en la ruta: SGSD/app/config

El objetivo de dicho fichero es realizar la configuración pertinente para cada uno de los servicios y categorías. Los parámetros que se han establecido como configurables son:

- Los servicios (*servicios*)
- Los buzones del servicio (*buzones*)
- Las categorías del servicio (*categorias*), las cuales a sus vez se configuran con los siguientes parámetros:
  - Alarma (*alarma*), tiene dos parámetros de configuración:
    - Tiempo en minutos hasta que salte la alarma (*max_time*)
    - Buzones excluidos de alarmas (*exclude_buzones*)
  - Prioridad (*prioridad*), se añaden los valores deseados por los que se considera el ticket prioritario(separados por '-' y saltos de línea).
  - Campos (*campos*), se añaden los campos que se desean mostrar en el monitor web, el primer carácter es indistinto en minúsculas o mayúsculas, si el campo no existe se lanzará un mensaje de error. Los campos se muestran separados por '-' y saltos de línea.

— Condiciones, (*condiciones*), se añaden las condiciones necesarias para cada categoría, formada por 3 campos (separados por '-' y saltos de línea):

- Campo relativo a la BBDD (*campo*), se indica el campo que se quiere comparar.
- Valor (*valor*), se indica el valor que se quiere comparar.
- Operación a realizar (*operacion*), se indica la operación que se quiere realizar entre el valor y el campo, por ejemplo, '=', 'LIKE', '>', etc.

  En la elaboración de la configuración se consideró que cada registro se interpreta como una operación lógica AND, posteriormente al comentarse la posible incursión de ciertos operadores lógicos 'OR' puntuales se ha añadido dicha configuración, siempre teniendo en cuenta que no se pueden realizar combinaciones complejas de AND y OR incluyendo paréntesis, sí se pueden hacer combinaciones simples de AND y OR.

  En caso de querer añadir a la condición la opción 'OR' habría que añadir '[OR]' en el campo *operación* a continuación del operador. Por ejemplo:

condiciones:
- { *campo*: estado, *valor*: OPEN, *operacion*: =[OR] }
- { *campo*: estado, *valor*: Work In progress, *operacion*: = }
- { *campo*: titulo, *valor*: %%iberia%%, *operacion*: NOT LIKE }

Esto sería equivalente en lenguaje SQL a:

(estado='OPEN' OR estado=' Work In progress' AND titulo NOT LIKE '%iberia%')

El carácter '%' es reservado y por lo tanto se debe escapar, para ello se introduce otro '%' previamente.

— Clientes críticos (*clientes_criticos*), se muestran separados por '-' y salto de línea aquellos clientes que se consideran críticos para la categoría.

— Días y horario en el que se muestran los tickets en caso de querer controlar su visionado (*intervalo_horario*). Se indica día y rango horario separados por '-' y saltos de línea. Si se quiere indicar que en un día no pueden visionarse los tickets dicho día no se inserta

- Día de la semana ,'lunes', 'martes',etc.. (*dia*)
- Hora desde la que se puede visionar(*desde*), expresada en formato 24 horas.
- Hora hasta la que se puede visionar(*hasta*), expresada en formato 24 horas.

Ejemplo:

intervalo_horario:
- { *dia*: jueves, desde: 08:00, *hasta*: 24:00 }
- { *dia*: viernes, desde: 08:00, *hasta*: 22:00 }

Hora menor 00:00
Hora mayor 24:00

- En el ejemplo expuesto solo se mostrarían tickets en los días y hora establecidos, es decir jueves y viernes.

- El intervalo de tiempo en segundos por el que se actualiza automáticamente la web (*actualiza_web_segundos*)

El esquema y sintaxis para definir los parámetros de configuración es el siguiente:

```
servicios:
  [nombre del servicio] (SOC, OIT, REHABITIC, etc)
  buzones:
  categorias:
        [nombre del ticket] (TICKETS OPEN, TICKETS FICTICIO, etc)
            alarma:
                max_time:
                exclude_buzones:
            prioridad:
            campos:
            condiciones:
                - { campo:  , valor:  , operacion:  }
            clientes_criticos:
            intervalo_horario:
                - { dia:  , desde:  , hasta:  }
```

Los servicios que tengan parámetros que no necesiten configurarse no se añaden, a continuación se muestran algunos ejemplos:

```
pi2_frac_sgsd_web_monitor:
    servicios:
        SOC:
            buzones:
                - SOC SEGURIDAD
                - SOPORTE SEGURIDAD
                - SEGEST MON
                - SEGEST SOC
                - SEGEST SEGUIMIENTO
                - SEGEST MONOCLIENTE
                - SSG CLIENTES
            categorias:
               'TICKETS OPEN':
                  alarma:
                     max_time: 15
                     exclude_buzones:
                        - SEGEST SOC
                        - SEGEST MON
                  prioridad:
                     - CRITICA
                     - ALTA
                  campos:
                     - NumeroCaso
                     - Prioridad
                     - TipoCaso
                     - GrupoDestino
                     - FechaActualizacion
                     - Titulo
                     - TecnicoAsignadoFinal
                  condiciones:
                     - { campo: estado, valor: 'OPEN', operacion: = }

            'TICKETS FICTICIO':
               prioridad:
                  - CRITICA
                  - ALTA
               campos:
                  - NumeroCaso
                  - Prioridad
                  - TipoCaso
                  - GrupoDestino
                  - FechaActualizacion
                  - Titulo
               condiciones:
                  - { campo: UsuarioAfectado, valor: FICTICIO, operacion: = }
               intervalo_horario:
                  - { dia: lunes, desde: 08:00, hasta: 24:00 }
                  - { dia: martes, desde: 08:00, hasta: 24:00 }
                  - { dia: miercoles, desde: 08:00, hasta: 24:00 }
                  - { dia: jueves, desde: 08:00, hasta: 24:00 }
                  - { dia: viernes, desde: 08:00, hasta: 22:00 }
```

Configuración Servicio OIT:

```
OIT:
    buzones:
        - SDNIVEL1
        - SDNIVEL1 SISTEVOZ
    categorias:
       'TICKETS OPEN':
          alarma:
             max_time: 15
          prioridad:
             - CRITICA
             - ALTA
          campos:
             - NumeroCaso
             - Prioridad
             - TipoCaso
             - GrupoDestino
             - FechaActualizacion
             - Titulo
             - TecnicoAsignadoFinal
             - Estado
          condiciones:
             - { campo: estado, valor: OPEN, operacion: =[OR] }
             - { campo: estado, valor: Work In progress, operacion: = }
             - { campo: Titulo, valor: %%iberia%%, operacion: not like }
          clientes_criticos:
             - VOCENTO
             - FERROVIAL
             - DGOG
             - SAICA
             - HOTELBEDS
             - NH
             - PRISA
             - DGPE
             - CEPSA
             - ICEX
             - GOBEX
             - NUTREXPA
```

Procedimiento almacenado

Se ha creado un procedimiento almacenado para la eliminación de los tickets que lleven más de n días en la base de datos.

Para ejecutar dicho procedimiento se puede hacer desde la ruta http://mi-sitio.com/phpmyadmin y en la pestaña sql poner la instrucción de llamada al procedimiento:

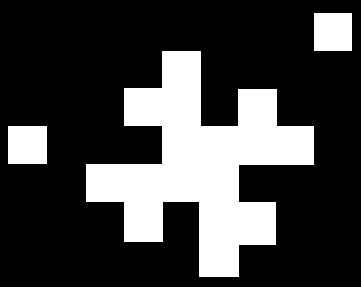
call eliminar_tickets(ndias)

en dias indicar el número de días a partir del cual queremos eliminar los tickets.

Ejemplo:

```
call eliminar_tickets(20)
Se eliminaran los tickets que lleven más o igual de 20 días en la base de datos
```

Manual de usuario SGSD



FRACTALIA