

# Anexo IV

## Modelo de Pruebas del Sistema

En este anexo, se muestra el plan de pruebas del sistema, junto con algunas estadísticas o datos de desempeño obtenidos utilizando una popular herramienta de “benchmarking”.

### 1. Plan de Pruebas

#### 1.1. Pruebas de unidad

Este tipo de pruebas se realizó especialmente sobre los componentes de acceso a la base de datos, o sea, las clases de diseño que están contenidas dentro del paquete de datos de la capa lógica del sistema. Estas clases de diseño, que son al fin de cuentas módulos de perl en la implementación de Thëwala son base esencial del sistema. Cada una de estas clases se probó por medio de guiones de prueba que realizaban diferentes acciones en la base de datos. A continuación se muestra un pequeño código de ejemplo que sirve para probar el módulo que maneja la tabla 'ps' de la base de datos. Se ha tratado de que los comentarios del código sean lo suficientemente explicativos:

```
#!/usr/bin/perl -w
# Guión de prueba de los módulos DBI
use strict;
use data::ps::PS;

# Se crea un Producto de nombre "Prod 3" y que pertenece a la empresa que tiene
# como id el número 2.
my $ps = data::ps::PS->create({
    name => "Prod1",
    company_id => 2,
});

# Se agrega el (los) producto(s) con nombre Prod3 al sector que tiene el id 5
foreach my $ps ( data::ps::PS->search( {name => "Prod3"} ) ) {
    $ps->add_to_sectors( {sector_id => 5} );
}

# Se buscan los productos que tengan la cadena "Prod" en su nombre
```

```

my $it = data::ps::PS->search_like( {name => "%Prod%"} );
# Se obtiene un "iterador" y para cada uno de los productos obtenidos en la
# búsqueda se ejecutan ciertas acciones
while(my $p = $it->next){
    # Se imprime el nombre
    print "found: ".$p->name."\n";
    foreach ($p->sectors()){
        # Se imprimen los ids de cada uno de los sectores
        print "\t ".$_->id."\n";
        # Si el id es 5 el producto se borra de la BD
        if ($_ == 5) { $_->delete();}
    };
};
}

```

## 1.2. Pruebas de subsistema

Las pruebas de subsistema se enfocaron principalmente en probar las colecciones de objetos de negocio expuestas como servicios web. Cada una de las colecciones de objetos fue probada asumiendo que los objetos de negocio funcionaban correctamente, de acuerdo a los resultados obtenidos en las pruebas de unidad. Para llevar a cabo las pruebas de subsistema se hizo necesario utilizar el servidor web Apache, junto con el módulo mod\_perl para exponer las colecciones de objetos como servicios web. Una vez estos servicios estuvieron expuestos, se utilizaron guiones de prueba para consumirlos remotamente. A continuación se muestra el código de un guión que llama a los métodos expuestos por el servicio web "Colección de Sectores", expuesto como SectorCollection en el servidor web. Se ha tratado de que los comentarios del código sean lo suficientemente explicativos:

```

#!/usr/bin/perl -w
# Guión de prueba para llamada a métodos remotos
# en el Servicio Web "SectorCollection"

use strict;

# Se usa la opción +trace=>['all'] para habilitar
# la depuración de todos los eventos de SOAP::Lite
use SOAP::Lite +trace=>['all'];

# Se crea un objeto SOAP::Lite para consumir el
# servicio web expuesto. Como proxy se utiliza la
# ubicación del guión de manejo del servicio web
# servido por Apache y como uri el identificador
# del servicio expuesto.
my $soap = SOAP::Lite

```

```

->proxy('http://localhost/ws/thewala/SectorCollection')
->uri('http://localhost/data/ps/SectorCollection')
->readable(1);

# Se crea un objeto SOAP con la estructura de un
# objeto "de tipo Sector".
my $sector = new SOAP::Data
    ->name('data_ps__Sector')
    ->value(bless { id => '0', name => 'otro sector'}, 'data_ps__Sector');

# Se agrega el objeto SOAP a la colección
# (servicio web) por medio del objeto SOAP::Lite.
my $lastkey = $soap->add($sector)->result;
# Se imprime el id de la ultima entrada de la db.
print "Last key: ".$lastkey."\n\n";

# Por medio del mismo objeto SOAP::Lite, se obtienen
# todos los sectores de la colección y se imprime su
# id y su nombre.
my $ret = $soap->getAll();
print "Sectors list:\n";
for my $s ($ret->valueof('//getAllResponse/data_ps__Sector')) {
    print $s->{id}." - ".$s->{name}."\n";
}

```

### 1.3. Pruebas de sistema

Las pruebas de sistema se basan principalmente en los casos de uso identificados. Por cada caso de uso, se debe completar el flujo normal de los eventos y cada uno de los flujos alternos y verificar que los resultados son los deseados. Las pruebas de sistema se realizaron a nivel de caja negra.

## 2. Datos de desempeño

Un tipo de pruebas interesantes, son las que tienen como objetivo medir la capacidad de respuesta o el desempeño de un sistema. Aunque dentro del proyecto Thèwala existen múltiples variables que pueden ser medidas, dentro de las pruebas del sistema sólo se tuvo en cuenta el servidor Web para realizar las pruebas de desempeño.

Para tener una idea del desempeño del servidor web cuando se invoca un guión de perl que hace llamados remotos a servicios web y luego muestra páginas web con el contenido de las respuestas, se utilizó la herramienta "ab" (Apache Benchmarking Tool), provista por Apache.

Esta herramienta se especializa en mostrar la cantidad de peticiones HTTP que

puede servir una máquina con Apache, por una determinada unidad de tiempo.

Las pruebas realizadas consistieron en utilizar "ab" para muestrear la capacidad de manejo de peticiones de apache, variando el número de peticiones concurrentes que podía manejar durante 30 segundos. El número de peticiones concurrentes se varió de 5 en 5, desde 1 hasta 35, ya que con este último número de peticiones concurrentes, la máquina con el servidor se bloqueó.

La máquina utilizada para las pruebas tenía un procesador de 1.86MHz, memoria RAM de 512MB, y sistema operativo Ubuntu Linux v5.4.

Los comandos de "ab" utilizados fueron los siguientes:

```
# ab -t 30 -c 5 -v 1 http://gorgona/thewala/control/ps/PSControl.pl?act=1
# ab -t 30 -c 10 -v 1 http://gorgona/thewala/control/ps/PSControl.pl?act=1
... y así sucesivamente.
```

La información más relevante de los resultados obtenidos se muestra a continuación:

```
Server Software:      Apache/1.3.33
Server Port:          80
Document Path:        /thewala/control/ps/PSControl.pl?act=1
-----
Concurrency Level:    1
-----
Time taken for tests:  30.189876 seconds
Complete requests:    77
Failed requests:      0
Write errors:         0
Requests per second:  2.55 [#/sec] (mean)
Time per request:     392.076 [ms] (mean)

Percentage of the requests served within a certain time (ms)
 50%    373
 66%    374
 75%    375
 80%    376
 90%    380
 95%    382
 98%    678
 99%   1028
100%   1028 (longest request)
-----
Concurrency Level:    5
-----
Time taken for tests:  30.66646 seconds
Complete requests:    62
Failed requests:      0
Write errors:         0
Requests per second:  2.06 [#/sec] (mean)
Time per request:     2424.730 [ms] (mean)

Percentage of the requests served within a certain time (ms)
 50%    2217
 66%    2441
 75%    2674
 80%    2742
 90%    3170
 95%    3738
 98%    5071
 99%    5509
100%    5509 (longest request)
```

```

-----
Concurrency Level:      10
-----
Time taken for tests:   30.117391 seconds
Complete requests:     60
Failed requests:       0
Write errors:          0
Requests per second:   1.99 [#/sec] (mean)
Time per request:      5019.565 [ms] (mean)

Percentage of the requests served within a certain time (ms)
 50%    4563
 66%    5169
 75%    5728
 80%    6012
 90%    7234
 95%    8190
 98%    8856
 99%    9576
100%   9576 (longest request)
-----
Concurrency Level:      15
-----
Time taken for tests:   30.226719 seconds
Complete requests:     47
Failed requests:       0
Write errors:          0
Requests per second:   1.55 [#/sec] (mean)
Time per request:      9646.825 [ms] (mean)

Percentage of the requests served within a certain time (ms)
 50%    8397
 66%    9523
 75%   10581
 80%   11868
 90%   14011
 95%   14936
 98%   15606
 99%   15606
100%  15606 (longest request)
-----
Concurrency Level:      20
-----
Time taken for tests:   30.463265 seconds
Complete requests:     46
Failed requests:       0
Write errors:          0
Requests per second:   1.51 [#/sec] (mean)
Time per request:      13244.898 [ms] (mean)

Percentage of the requests served within a certain time (ms)
 50%   10559
 66%   12470
 75%   13293
 80%   15445
 90%   16342
 95%   17620
 98%   27147
 99%   27147
100%  27147 (longest request)
-----
Concurrency Level:      25
-----

```

```
Time taken for tests: 30.386300 seconds
Complete requests: 33
Failed requests: 0
Write errors: 0
Requests per second: 1.09 [#/sec] (mean)
Time per request: 23019.924 [ms] (mean)

Percentage of the requests served within a certain time (ms)
 50% 15445
 66% 17734
 75% 17922
 80% 20545
 90% 23999
 95% 26662
 98% 27803
 99% 27803
100% 27803 (longest request)
-----

Concurrency Level: 30
-----

Time taken for tests: 30.694831 seconds
Complete requests: 38
Failed requests: 0
Write errors: 0
Requests per second: 1.24 [#/sec] (mean)
Time per request: 24232.762 [ms] (mean)

Percentage of the requests served within a certain time (ms)
 50% 18243
 66% 21239
 75% 22234
 80% 23249
 90% 24443
 95% 24984
 98% 29657
 99% 29657
100% 29657 (longest request)
-----

Concurrency Level: 35
-----

El servidor se bloqueó con este número de peticiones concurrentes
```

Luego de esto se realizó una prueba consistente en realizar 5 medidas de 30 segundos, con 5 peticiones concurrentes cada una. La información más relevante de los resultados obtenidos se muestra a continuación:

```
-----
Concurrency Level: 5
-----

Time taken for tests: 30.27668 seconds
Complete requests: 53
Failed requests: 0
Write errors: 0
Requests per second: 1.77 [#/sec] (mean)
Time per request: 2832.799 [ms] (mean)

Percentage of the requests served within a certain time (ms)
 50% 2271
 66% 2623
 75% 2899
 80% 3036
 90% 4665
 95% 8379
 98% 8894
 99% 9200
100% 9200 (longest request)
```

```

-----
Concurrency Level:      5
-----
Time taken for tests:   30.185662 seconds
Complete requests:     68
Failed requests:       0
Write errors:          0
Requests per second:   2.25 [#/sec] (mean)
Time per request:      2219.534 [ms] (mean)

Percentage of the requests served within a certain time (ms)
 50%    1776
 66%    2231
 75%    2830
 80%    2938
 90%    3447
 95%    3939
 98%    4487
 99%    4621
100%    4621 (longest request)
-----

-----
Concurrency Level:      5
-----
Time taken for tests:   30.214182 seconds
Complete requests:     67
Failed requests:       0
Write errors:          0
Requests per second:   2.22 [#/sec] (mean)
Time per request:      2254.790 [ms] (mean)

Percentage of the requests served within a certain time (ms)
 50%    2064
 66%    2443
 75%    2601
 80%    2719
 90%    3465
 95%    3859
 98%    4323
 99%    5105
100%    5105 (longest request)
-----

-----
Concurrency Level:      5
-----
Time taken for tests:   30.38276 seconds
Complete requests:     72
Failed requests:       0
Write errors:          0
Requests per second:   2.40 [#/sec] (mean)
Time per request:      2085.991 [ms] (mean)

Percentage of the requests served within a certain time (ms)
 50%    1926
 66%    2294
 75%    2517
 80%    2560
 90%    2865
 95%    3133
 98%    3244
 99%    3595
100%    3595 (longest request)
-----

```

Luego de esto se realizó una prueba consistente en realizar una medida de 5 minutos de duración, con 5 peticiones concurrentes. La información más relevante de los resultados obtenidos se muestra a continuación:

```
-----  
Concurrency Level:      5  
-----  
Time taken for tests:   300.265416 seconds  
Complete requests:     768  
Failed requests:       0  
Write errors:          0  
Requests per second:   2.56 [#/sec] (mean)  
Time per request:      1954.853 [ms] (mean)  
  
Percentage of the requests served within a certain time (ms)  
 50%    1862  
 66%    2116  
 75%    2305  
 80%    2437  
 90%    2745  
 95%    3015  
 98%    3361  
 99%    3577  
100%    4198 (longest request)  
-----
```

**THĒWALA: SISTEMA DE INFORMACIÓN DE  
PRODUCTOS Y SERVICIOS PARA LAS  
EMPRESAS PERTENECIENTES A LA  
CORPORACIÓN PARQUE  
TECNOLÓGICO DE POPAYÁN**

**Diego Andrés Asenjo González  
Alejandro Ríos Peña**

**Universidad del Cauca  
Facultad de Ingeniería Electrónica y Telecomunicaciones  
Departamento de Telemática  
Popayán, Colombia  
Mayo de 2.005**

**ANEXO IV**  
**MODELO DE PRUEBAS DEL SISTEMA**

**Diego Andrés Asenjo González**  
**Alejandro Ríos Peña**

Director

**Mario Fernando Solarte Sarasty**  
Magíster en Ingeniería Telemática

**Universidad del Cauca**  
**Facultad de Ingeniería Electrónica y Telecomunicaciones**  
**Departamento de Telemática**  
**Popayán, Colombia**  
**Mayo de 2.005**

## **CONTENIDO**

Modelo de pruebas del sistema	1
1. Plan de Pruebas	1
1.1. Pruebas de unidad	1
1.2. Pruebas de subsistema	2
1.3. Pruebas de sistema	3
2. Datos de desempeño	3