

Anexo 3 : Pruebas Amazon AWS

En este anexo se reflejaran de forma muy resumida las pruebas realizadas con Amazon AWS en sus dos servicios básicos EC2 y S3.

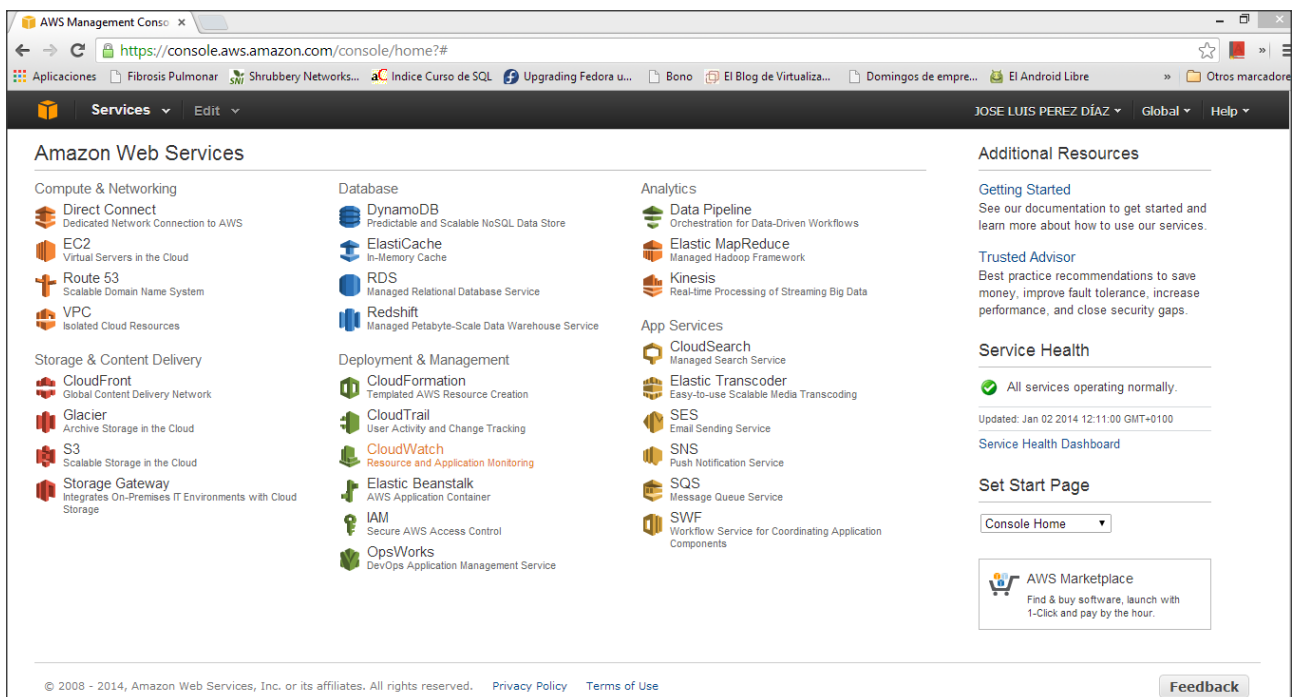
■ Creación de Cuenta Amazon AWS.

Se parte inicialmente creando una cuenta Amazon AWS. Es un proceso simple, sin complejidad alguna, que arranca en la página web <http://aws.amazon.com/es/console/>.

■ Login inicial.

En la fase de creación de cuenta se obtiene un login y un password que posteriormente se utilizarán para entrar en la consola de administración.

Una vez introducidos ambos y validados se tiene acceso a la consola donde se tiene acceso a todos los servicios disponibles.



En la esquina superior aparece el titular de la cuenta. Veremos posteriormente que haciendo click sobre éste se tendrá acceso a las funciones administrativas de la cuenta, como por ejemplo la facturación, etc...

Igualmente a la derecha aparece la etiqueta Global (que aunque en esta pantalla no es de uso), en las siguientes se utilizará para seleccionar la Zona Geográfica de uso de los servicios. Ello es particularmente importante por las implicaciones legales para Datos (según cada zona geográfica), Precios (no son iguales en todas las zonas), etc...

Nota: Cada vez que se seleccione un servicio, se ha de verificar o en su caso cambiar, a la zona deseada previa a la provisión del servicio.

▪ Servicio EC2

Al seleccionar la opción EC2 se entra directamente en la pantalla de gestión de Instancias. El interface es intuitivo y de fácil manejo. Presenta información resumida de los recursos consumidos, etc...

En mi caso concreto:

- 1 Instancias
- 1 Volumen
- 1 IP
- 1 Key Pair

Y Por debajo directamente tiene el enlace para "Launch Instance" que veremos posteriormente.

The screenshot shows the AWS Management Console for the EC2 service in the US West (Oregon) region. The 'Resources' section indicates the following usage:

1 Running Instance	1 Elastic IP
1 Volume	0 Snapshots
1 Key Pair	0 Load Balancers
0 Placement Groups	2 Security Groups

Below the resource summary, there is a 'Create Instance' section with a prominent 'Launch Instance' button. The 'Service Health' section shows that the US West (Oregon) service is operating normally. The 'Account Attributes' section on the right lists supported platforms, VPC, and additional information links.

En la siguiente pantalla tenemos información específica (al hacer click sobre Running Instance), de las instancias creadas.

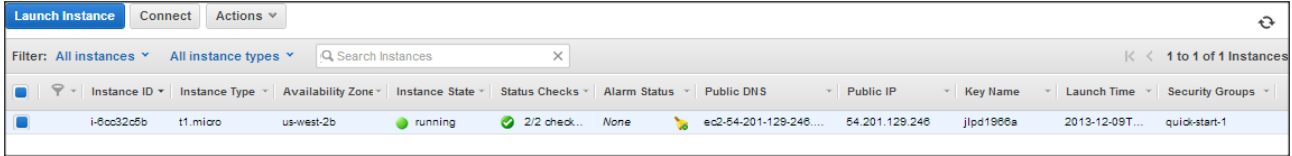
The screenshot displays the 'Instances' page in the AWS Management Console. A table lists the instances:

Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS	Public IP	Key Name	Launch Time	Security Groups
i-6cc32c5b	t1.micro	us-west-2b	running	2/2 checks...	None	ec2-54-201-129-246 u...	54.201.129.246	jlpd1966a	2013-12-09T0...	quick-start-1

At the bottom of the page, there is a footer with copyright information and a 'Feedback' button.

La ip que tiene asignada es la 54.201.129.246

Como en el caso de OpenStack, se debe elegir en el momento de la creación ,un tipo de VM, elegir una imagen, elegir unas reglas de seguridad,etc... y finalmente asignarle ip flotante y en su caso agregarle un volumen. Por otr parte los ciclos de vida de las instancias son exactamente igual que en el caso de OpenStack.



Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS	Public IP	Key Name	Launch Time	Security Groups
i-8cc32c5b	t1.micro	us-west-2b	running	2/2 check...	None	ec2-54-201-129-246...	54.201.129.246	jlpd1966a	2013-12-09T...	quick-start-1

En el caso que nos ocupa he utilizado una instancia t1.micro, con un SecurityGroup quick-start-1, y con una keypair jlpd1966a.

Una vez creada la instancia es accesible remotamente vía ssh haciendo uno de la clave. A continuación le haré un traceroute por constatar el path que sigue la conexión desde la instancia en la Zona elegida en Oregon. (efectivamente por los dominios se constata que viene desde Seattle,etc...)

```
[root@fc18-atica .ssh]# ssh -i jlpd1966a.pem ec2-user@54.201.129.246
Last login: Tue Dec 10 12:57:06 2013 from 212.230.253.254
```

```
__| __|_ )
_| ( / Amazon Linux AMI
__|\__|__|
```

<https://aws.amazon.com/amazon-linux-ami/2013.09-release-notes/>

```
[ec2-user@ip-172-31-31-141 ~]$ traceroute 212.231.5.85
```

traceroute to 212.231.5.85 (212.231.5.85), 30 hops max, 60 byte packets

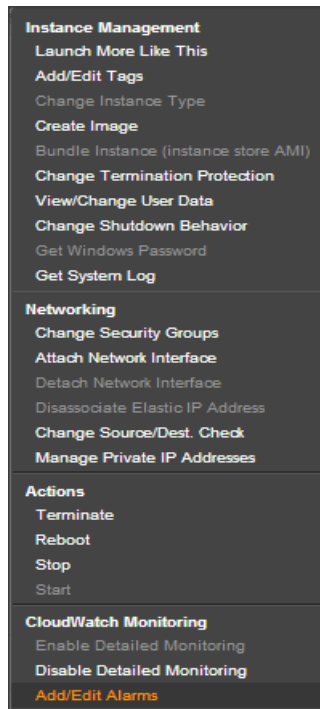
```
 1 ec2-50-112-0-170.us-west-2.compute.amazonaws.com (50.112.0.170) 0.988 ms 0.732 ms 0.718 ms
 2 205.251.232.166 (205.251.232.166) 4.546 ms 4.491 ms 4.544 ms
 3 205.251.232.148 (205.251.232.148) 3.401 ms 205.251.232.160 (205.251.232.160) 1.388 ms 1.370 ms
 4 205.251.232.89 (205.251.232.89) 20.407 ms 205.251.232.95 (205.251.232.95) 13.385 ms 205.251.232.93 (205.251.232.93)
 16.171 ms
 5 205.251.225.197 (205.251.225.197) 20.026 ms 205.251.225.165 (205.251.225.165) 13.348 ms 205.251.225.179
 (205.251.225.179) 13.255 ms
 6 sea-b1-link.telia.net (213.248.94.29) 16.190 ms sea-b1-link.telia.net (62.115.34.105) 19.982 ms sea-b1-link.telia.net
 (213.248.84.189) 13.136 ms
 7 level3-ic-304311-sea-b1.c.telia.net (62.115.37.118) 13.383 ms 13.345 ms 13.496 ms
 8 ae-32-52.ebr2.Seattle1.Level3.net (4.69.147.182) 176.357 ms 176.336 ms 176.302 ms
 9 ae-2-2.ebr2.Denver1.Level3.net (4.69.132.54) 175.757 ms 176.157 ms 176.101 ms
10 ae-3-3.ebr1.Chicago2.Level3.net (4.69.132.62) 175.910 ms 174.705 ms 174.648 ms
11 ae-6-6.ebr1.Chicago1.Level3.net (4.69.140.189) 175.098 ms 175.079 ms 175.047 ms
12 ae-2-2.ebr2.NewYork2.Level3.net (4.69.132.66) 176.360 ms 175.554 ms 175.929 ms
13 ae-1-100.ebr1.NewYork2.Level3.net (4.69.135.253) 176.075 ms 176.263 ms 176.236 ms
14 ae-4-4.ebr1.NewYork1.Level3.net (4.69.141.17) 175.540 ms ae-46-46.ebr1.NewYork1.Level3.net (4.69.201.41) 175.951 ms ae-48-
48.ebr2.NewYork1.Level3.net (4.69.201.49) 177.658 ms
15 ae-44-44.ebr2.London1.Level3.net (4.69.137.77) 175.273 ms 175.979 ms ae-42-42.ebr2.London1.Level3.net (4.69.137.69)
175.433 ms
16 vlan102.ebr1.London1.Level3.net (4.69.143.89) 175.507 ms 175.747 ms vlan103.ebr1.London1.Level3.net (4.69.143.93) 175.090
```

```

ms
17 ae-45-45.ebr1.Paris1.Level3.net (4.69.143.102) 176.313 ms ae-48-48.ebr1.Paris1.Level3.net (4.69.143.114) 175.566 ms ae-45-45.ebr1.Paris1.Level3.net (4.69.143.102) 175.779 ms
18 ae-71-71.csw2.Paris1.Level3.net (4.69.161.82) 175.791 ms 174.774 ms ae-61-61.csw1.Paris1.Level3.net (4.69.161.78) 175.411 ms
19 ae-63-63.ebr3.Paris1.Level3.net (4.69.161.109) 175.585 ms ae-83-83.ebr3.Paris1.Level3.net (4.69.161.117) 175.200 ms 174.677 ms
20 * * *
21 213.242.113.122 (213.242.113.122) 175.780 ms 175.300 ms 176.233 ms
22 212.230.7.9 (212.230.7.9) 177.526 ms 177.222 ms 176.959 ms
23 212.231.7.26 (212.231.7.26) 176.074 ms 176.815 ms 176.577 ms
24 212.231.5.85 (212.231.5.85) 177.711 ms 177.696 ms 177.671 ms
25 * * *
26 * * *
27 * * *
28 * * *
29 *^C
[ec2-user@ip-172-31-31-141 ~]$

```

El interface de Amazon AWS es sencillo y muy extenso para cubrir todo tipo de opciones cubriendo todos los aspectos de las instancias y sus recursos asociados.



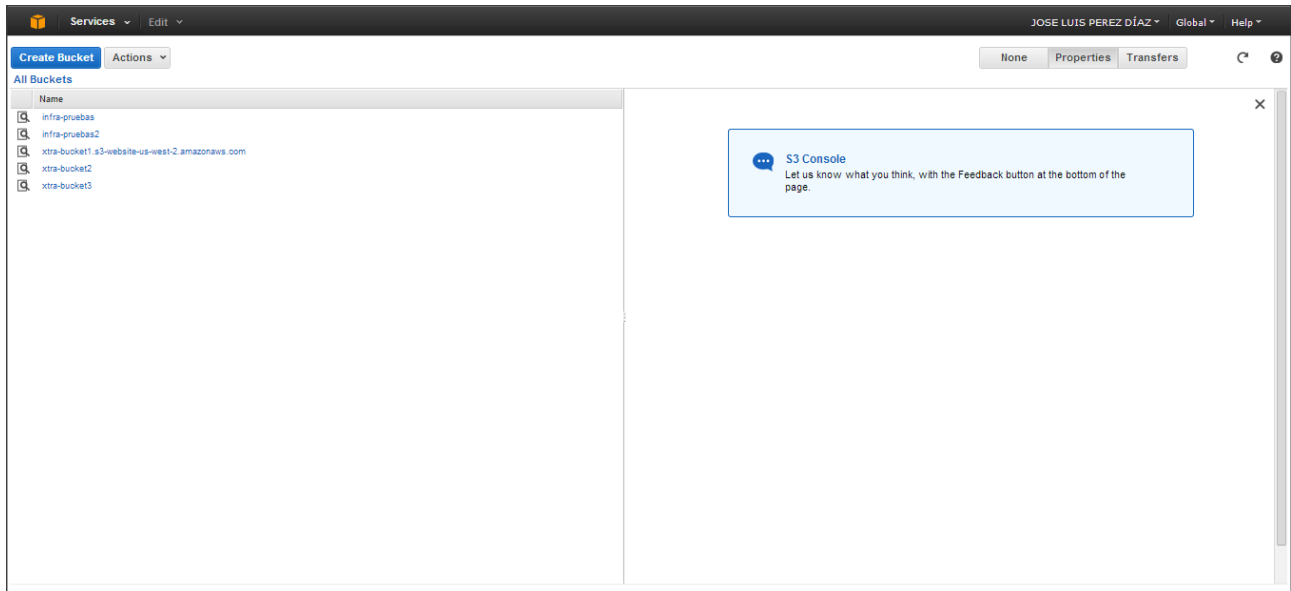
Finalmente mencionar que el interface relativo a los volúmenes es muy sencillo e intuitivo para asociar todo tipo de volúmenes a las instancias.

Name	Volume ID	Capacity	Volume Type	Snapshot	Created	Zone	State	Alarm Status	Attachment Information	Monitoring	Volume Status
empty	vol-02dc8f6b	8 GiB	standard	snap-4dc80b71	2013-10-01T13:10	us-west-2b	in-use	none	i-8cc32c5b:/dev/sda1 (att		Okay

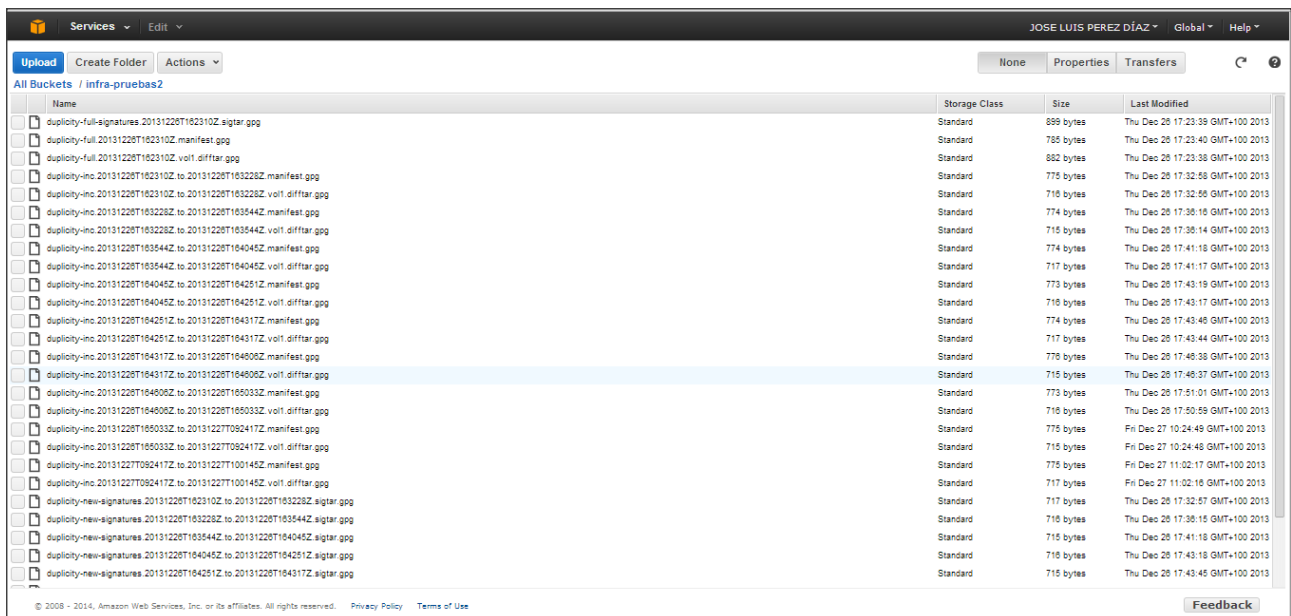
▪ Servicio S3

El servicio S3 se utiliza mayormente vía API, aunque hay numerosas herramientas disponibles (en las que al configurarlas) tienes pleno acceso a S3.

Básicamente todo arranca con la creación de Depósitos o Buckets en los que almacenas la información. Las pruebas se han realizado con el proyecto Opensource Duplicity.



Haciendo doble click en los bucket se ve la información relativa a cada depósito.



Es ideal el uso de S3 para aplicaciones de backups o contenido estático. Particularmente usamos la herramienta duplicity para hacer backups incrementales encriptados sobre S3.

Me apoye en la web del proyecto duplicity (<http://duplicity.nongnu.org/>) y en el caso practico planteado por Thomas Sileo en su blog (<http://thomassileo.com/blog/2012/07/19/ubuntu-slash-debian-encrypted-incremental-backups-with-duplicity-on-amazon-s3/>).

A continuación replico un script ejemplo para hacerlos backups. Requiere instalar las herramientas, generar las claves y configurar este script con los datos específicos de cada uno. Finalmente se podría lanzar con el cron,etc...

```
#!/bin/bash
export AWS_ACCESS_KEY_ID=YOUR_ACCESS_KEY
export AWS_SECRET_ACCESS_KEY=YOUR_SECRET_ACCESS_KEY
export PASSPHRASE=YOU_PASSHRASE

# directories, space separated
SOURCE="/home/thomas/backup /home/thomas/bin /home/thomas/documents"
BUCKET=s3+http://mybucket
LOGFILE=/home/thomas/tmp/duplicity.log
# set email to receive a backup report
EMAIL=""

backup() {
    INCLUDE=""
    for CDIR in $SOURCE
    do
        TMP="--include ${CDIR}"
        INCLUDE=${INCLUDE}${TMP}
    done
    # perform an incremental backup to root, include directories, exclude everything else, / as reference.
    duplicity --full-if-older-than 30D $INCLUDE --exclude '*' / $BUCKET > $LOGFILE
    if [ -n "$EMAIL" ]; then
        mail -s "backup report" $EMAIL < $LOGFILE
    fi
}

list() {
    duplicity list-current-files $BUCKET
}

restore() {
    if [ $# = 2 ]; then
        duplicity restore --file-to-restore $1 $BUCKET $2
    else
        duplicity restore --file-to-restore $1 --time $2 $BUCKET $3
    fi
}

status() {
    duplicity collection-status $BUCKET
}

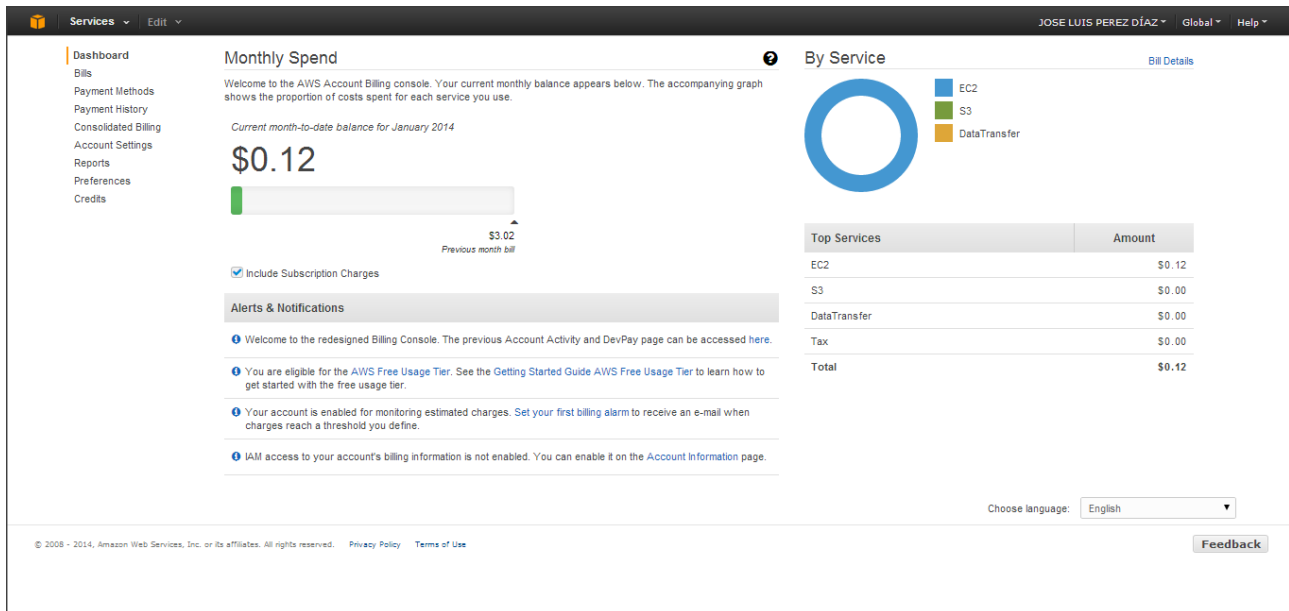
if [ "$1" = "backup" ]; then
    backup
elif [ "$1" = "list" ]; then
    list
elif [ "$1" = "restore" ]; then
    if [ $# = 3 ]; then
        restore $2 $3
    else
        restore $2 $3 $4
    fi
elif [ "$1" = "status" ]; then
    status
else
    echo "
duptools - manage duplicity backup

USAGE:
./duptools.sh backup
./duptools.sh list
./duptools.sh status
./duptools.sh restore file [time] dest
"
fi

export AWS_ACCESS_KEY_ID=
export AWS_SECRET_ACCESS_KEY=
export PASSPHRASE=
```

■ Facturación

Uno de los aspectos fundamentales de Amazon AWS es, además de los servicios en si mismos, la introducción de mecanismos de facturación online, con todo tipo de facilidades que son configurables y que permiten al Cliente evaluar los costes en los que incurre.



■ Conclusiones.

Sin lugar a dudas Amazon AWS es fantástico a todos los niveles. No ha sido el objeto de este Proyecto profundizar en sus muchos detalles, pero si introducir la alternativa que de hecho es, y que se debe considerar. Es el líder de referencia y lleva años de ventaja sobre el resto de los competidores/alternativas.

En lo que se refiere a EC2 y S3, son fáciles de usar, y muy efectivos. La información es muy abundante y el uso de ambos entornos en cuasi-inmediato.

No obstante lo dicho, los costes en lo que se podría incurrir y para cierto tipo de clientes, pueden justificar la inversión en Infraestructura propia. En este caso OpenStack es una alternativa similar y con un proyecto detrás que prometer evolucionar para ser cada año más competitivo.