



# INFORME PROYECTO FINAL

Microblogging en la Nube

Computación Distribuida

Ing. Iván Carrera

Stalin Guamán, David Tayupanta

## Contenido

1.	Alcance del proyecto.....	2
2.	Diseño de la aplicación en la nube.....	3
3.	Implementación.....	3
3.1.	Algoritmo .....	3
3.2.	Código fuente.....	3
4.	Pruebas de Funcionamiento (Ejecución) .....	4
4.1.	Panel Microsoft Azure.....	4
4.2.	Clientes.....	4
4.3.	Balanceador.....	6
4.4.	Servidores.....	6
4.4.1.	Servidor de Respaldo, en caso de caída.....	7
4.5.	Base de Datos.....	9
4.6.	Publicación Web.....	9
5.	Conclusiones y trabajo futuro .....	11

## 1. Alcance del proyecto

El proyecto a realizarse consiste en un Servicio de Microblogging en donde un número de N clientes accede al servicio enviando mensajes por medio de un balanceador front end, el cual distribuye los mensajes recibidos por el cliente a los servidores correspondientes. Para posteriormente publicar los mensajes recibidos en una página web.

Los clientes pueden acceder a través de una aplicación de escritorio al servicio y en ella pueden ingresar mensajes (Ejemplo: hola soy el nuevo cliente) y una ubicación geográfica (Ejemplo: Quito). Este mensaje será receptado por el Balanceador de carga y dependiendo de la ubicación geográfica del mensaje, que en el ejemplo sería Quito, este mensaje será enviado al servidor de Quito para que realice los procesos correspondientes.

Finalmente todos los mensajes, indistintamente en qué servidor se encuentren, será publicado en una página web. Mientras reciba nuevos mensajes, los mensajes anteriores se colocaran en la parte inferior continuamente y los mensajes nuevos en la parte superior (similar a una pila de datos). En esta página web se mostrará el mensaje del cliente y la ubicación geográfica (Ejemplo: hola soy el nuevo cliente - Quito). Si uno de los servidores deja de funcionar la ciudad involucrada inmediatamente se quede sin el servicio. El servicio será funcional únicamente para tres ciudades que son Quito, Guayaquil y Cuenca.

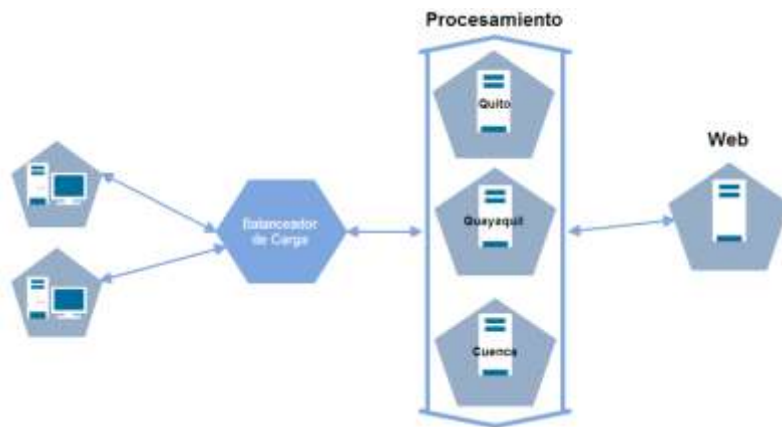
Pensemos en esta aplicación aplicada a un entorno laboral en el cual los empleados pondrán a consideración sus necesidades y novedades.

No muy distinto de lo que sería un muro de una red social pero a su vez con una gran ventaja que es la eliminación de las distracciones externas que puede generar las publicaciones ajenas a lo laboral que pueden ser un factor de distracción en el proceso de comunicación de una compañía.

Así pues muro pichones puede contribuir enormemente con la eliminación de distracciones a la hora de trabajar.

Pero no solo transformaremos a muro pichones en una herramienta empresarial únicamente sino más bien lo que queremos hacer de esta es una aplicación que con unas pequeñas modificaciones pueda tomar características propias de una corporación o de un equipo de trabajo generando un espíritu de cariño hacia la institución para la cual muro pichones prestará sus servicios.

## 2. Diseño de la aplicación en la nube



## 3. Implementación

### 3.1. Algoritmo

```

Procedure EscritorioAplicacion{
  Ingresar Mensaje;
  Seleccionar Lugar de envío
  Enviar Parametros
}

Procedure Balanceador {
  Recibe Parametros
  Balanceador de Carga {
    Switch(Lugar de envío)
      case 1: enviar a servidor en Quito;
      case 2: enviar a servidor en Guayaquil;
      case 3: enviar a servidor en Cuenca;
  }
}

Procedure Servidor Quito , Guayaquil, Cuenca{
  leer mensaje
  enviar publicación
}

Servidor WS{
  recibir publicación
  cargar publicación
}
  
```

### 3.2. Código fuente

El código fuente se lo puede encontrar en el siguiente link:

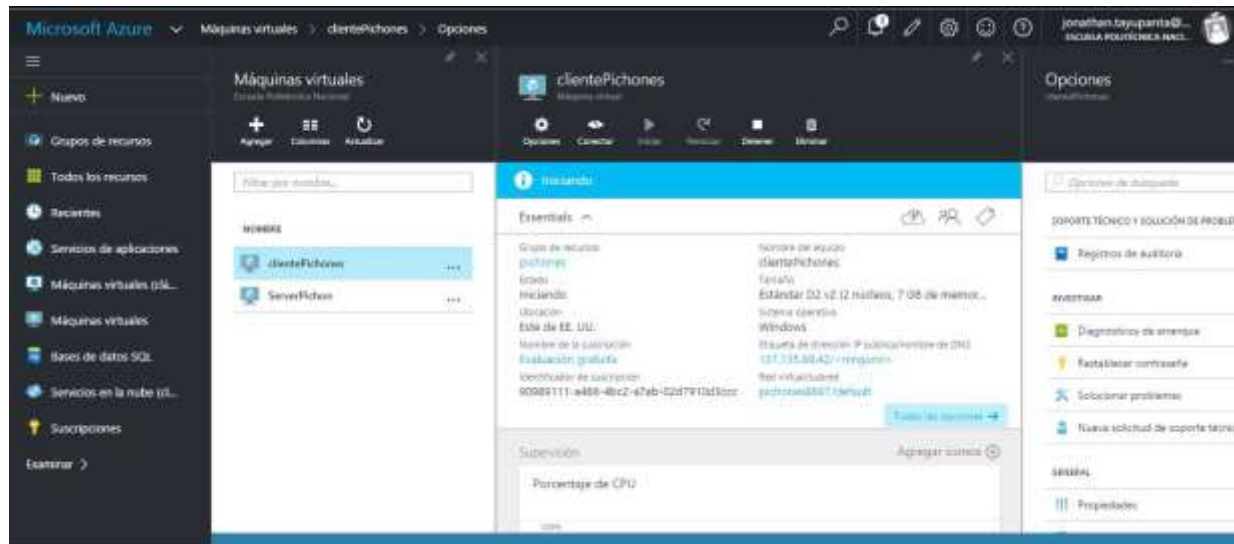
<https://github.com/StanGumn/proyectoMicrobloggingDistribudas>

## 4. Pruebas de Funcionamiento (Ejecución)

### 4.1. Panel Microsoft Azure

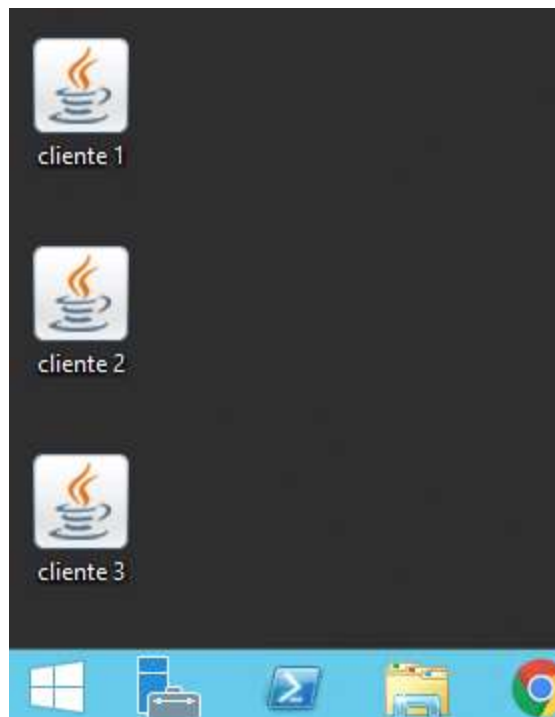
Aquí mostramos el panel de microsoft Azure en la cual se muestra la arquitectura empleada para el desarrollo de nuestro proyecto.

Para la ejecución de nuestra aplicación encendemos las computadoras que contienen los distintos componentes que intervienen en la ejecución de la aplicación.



### 4.2. Clientes

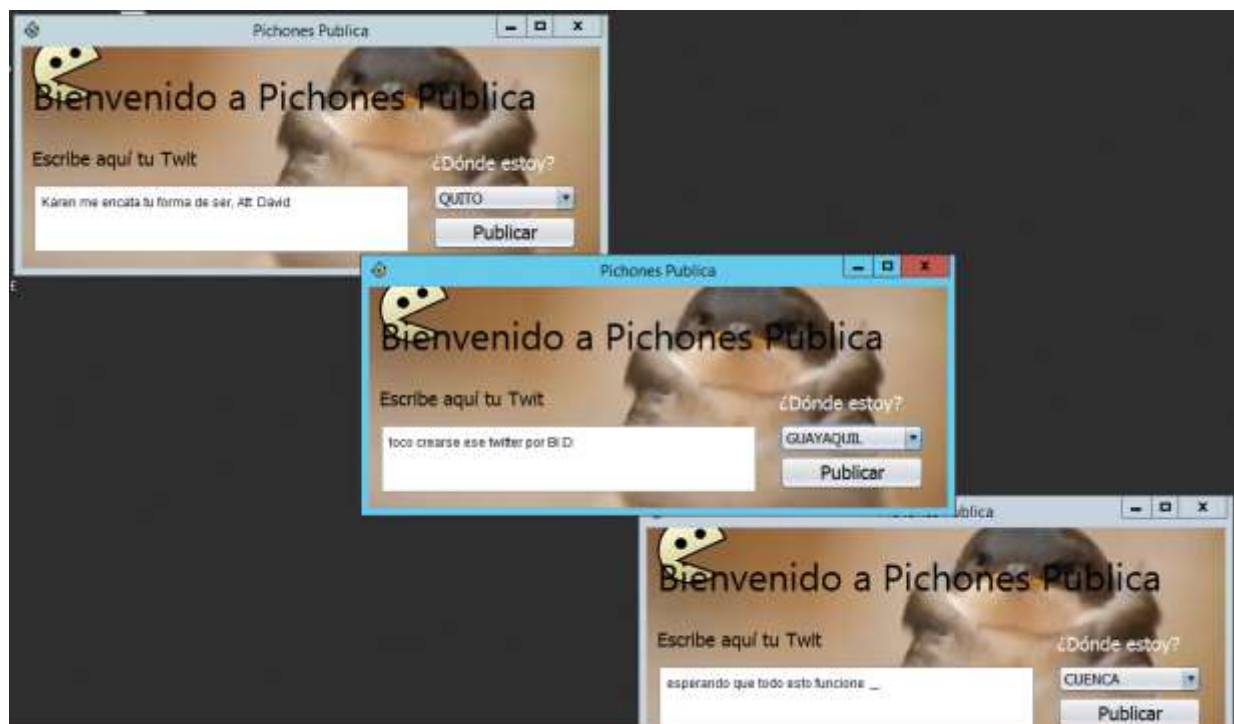
A continuación empezamos con la ejecución de los clientes mediante su interfaz gráfica como se observa en la figura a continuación:



Se puede ejecutar N clientes, pero para este ejemplo práctico tenemos tres clientes. La ejecución de cada uno de ellos las podemos apreciar a continuación



Dentro de cada uno de ellos se puede escribir la publicación y seleccionar la ubicación geográfica en la que se encuentra (Quito, Guayaquil y Cuenca), esto hará que los mensajes se dirijan a los servidores correspondientes a cada ciudad. A continuación podemos apreciar los mensajes que serán publicados



Para finalizar la ejecución de la aplicación de cliente, podemos observar a continuación un mensaje de confirmación



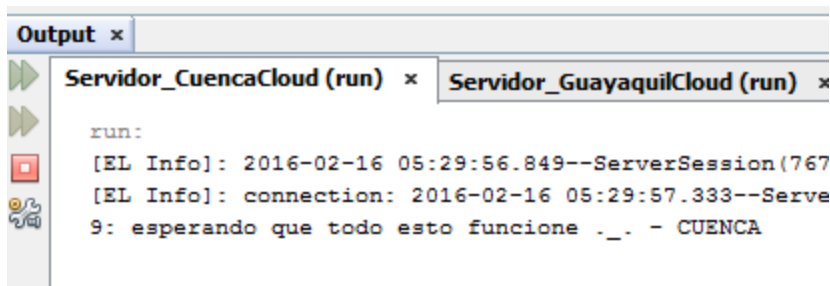
Con esto los mensajes se han enviado correctamente a los servidores para su futura publicación, antes sin olvidarnos que los mensajes pasaran por el balanceador que observaremos a continuación en el siguiente punto.

#### 4.3. Balanceador



#### 4.4. Servidores

Para continuar con el ejemplo, observaremos el funcionamiento adecuado de los tres servidores, para ello tenemos el primer servidor de cuenca a continuación con el respectivo mensaje que fue enviado a dicha ciudad:



Ahora la ciudad de Guayaquil

```

Servidor_CuencaCloud (run) x Servidor_GuayaquilCloud (run) x
run:
[EL Info]: 2016-02-16 05:29:54.302--ServerSession(780652
[EL Info]: connection: 2016-02-16 05:29:54.755--ServerSe
9: toco crearse ese twitter por BI D: - GUAYAQUIL

```

Y finalmente la ciudad de Quito

```

Servidor_CuencaCloud (run) x Servidor_GuayaquilCloud (run) x Servidor_QuitoCloud (run) x
run:
[EL Info]: 2016-02-16 05:29:50.357--ServerSession(1115332350)--EclipseLink, vers
[EL Info]: connection: 2016-02-16 05:29:50.826--ServerSession(1115332350)--file:
9: Karen me encata tu forma de ser, Att: David - QUITO

```

Con esto probamos el funcionamiento adecuado de los servidores, pero ¿qué pasa en caso de que se caiga cualquiera de ellos? Para este caso tenemos un servidor de respaldo que observaremos a continuación

#### 4.4.1. Servidor de Respaldo, en caso de caída

En caso de que uno o más servidores se caigan, el servidor de respaldo entra en acción y se ejecuta mientras se logra solucionar los problemas encontrados en los servidores caídos.

Para este ejemplo detenemos la ejecución del servidor de Quito y Guayaquil quedando en funcionamiento únicamente el de Cuenca y el Respaldo.

```

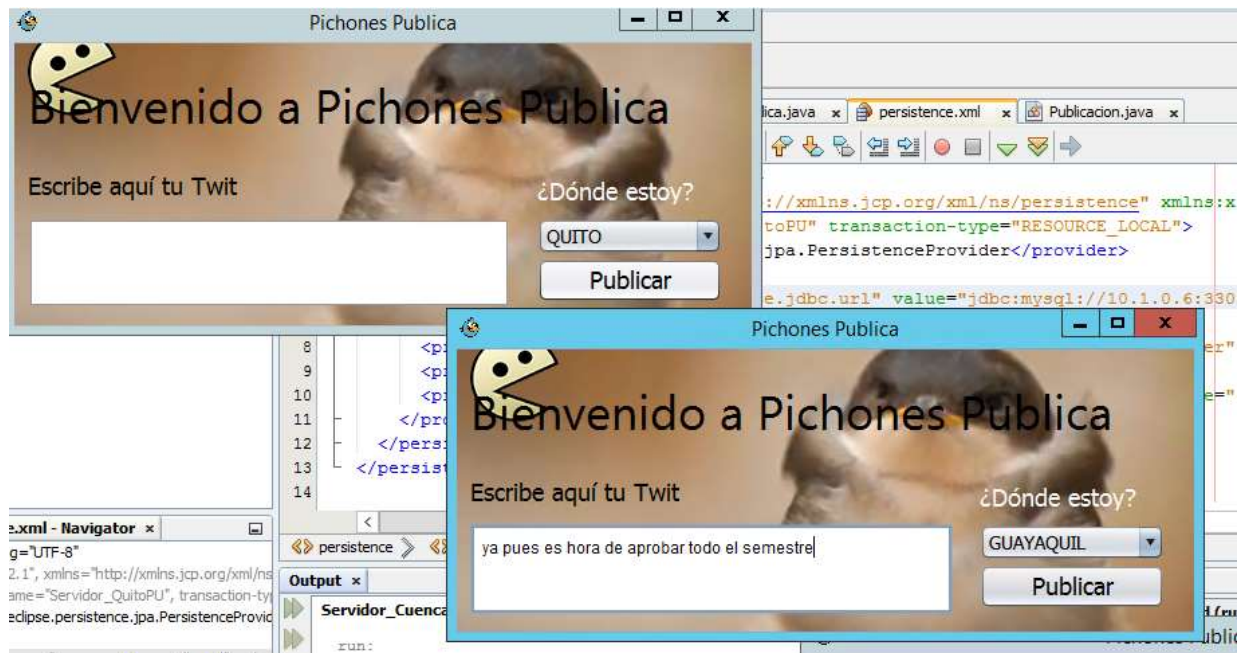
Servidor_CuencaCloud (run) x Servidor_GuayaquilCloud (run) x Servidor_QuitoCloud (run) x
run:
[EL Info]: 2016-02-16 05:29:50.357--ServerSession(1115332350)--EclipseLi
[EL Info]: connection: 2016-02-16 05:29:50.826--ServerSession(1115332350
9: Karen me encata tu forma de ser, Att: David - QUITO

BUILD STOPPED (total time: 27 minutes 33 seconds)

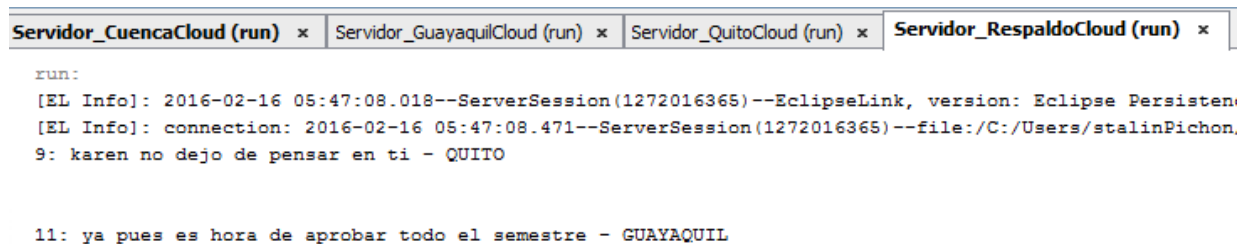
```

Ahora ejecutamos los clientes enviando mensajes a estos servidores





Ya que no se encuentran en funcionamiento los servidores de Guayaquil y Quito, el servidor de respaldo es quien nos ayuda a solucionar la disponibilidad de nuestro sistema. Como podemos apreciar en la siguiente figura, podemos ver el funcionamiento correcto del servidor de Respaldo



Para ello el balanceador mostrará un mensaje en donde indica que servidor no se encuentra en funcionamiento, esto lo podemos observar en la siguiente imagen.

```

Output x
Balanceador (run) x muroPichoness (run) x Java DB Database Process x GlassFish Server x

RUN:
Servidor quito funcionando
Gracias por usar Pichones Publica
Servidor Guayaquil funcionando
Gracias por usar Pichones Publica
Servidor Cuenca funcionando
Gracias por usar Pichones Publica
Couldn't get I/O for the connection to 10.1.0.7
el servidor de Quito dejo de funcionar solucionar problema
Servidor alterno funcionando
Gracias por usar Pichones Publica
Couldn't get I/O for the connection to 10.1.0.7
el servidor de Guayaquil dejo de funcionar solucionar problema
Servidor alterno funcionando
Gracias por usar Pichones Publica
  
```

#### 4.5. Base de Datos

Aquí comprobamos que efectivamente nuestra base de datos se encuentra almacenando cada una de las publicaciones.

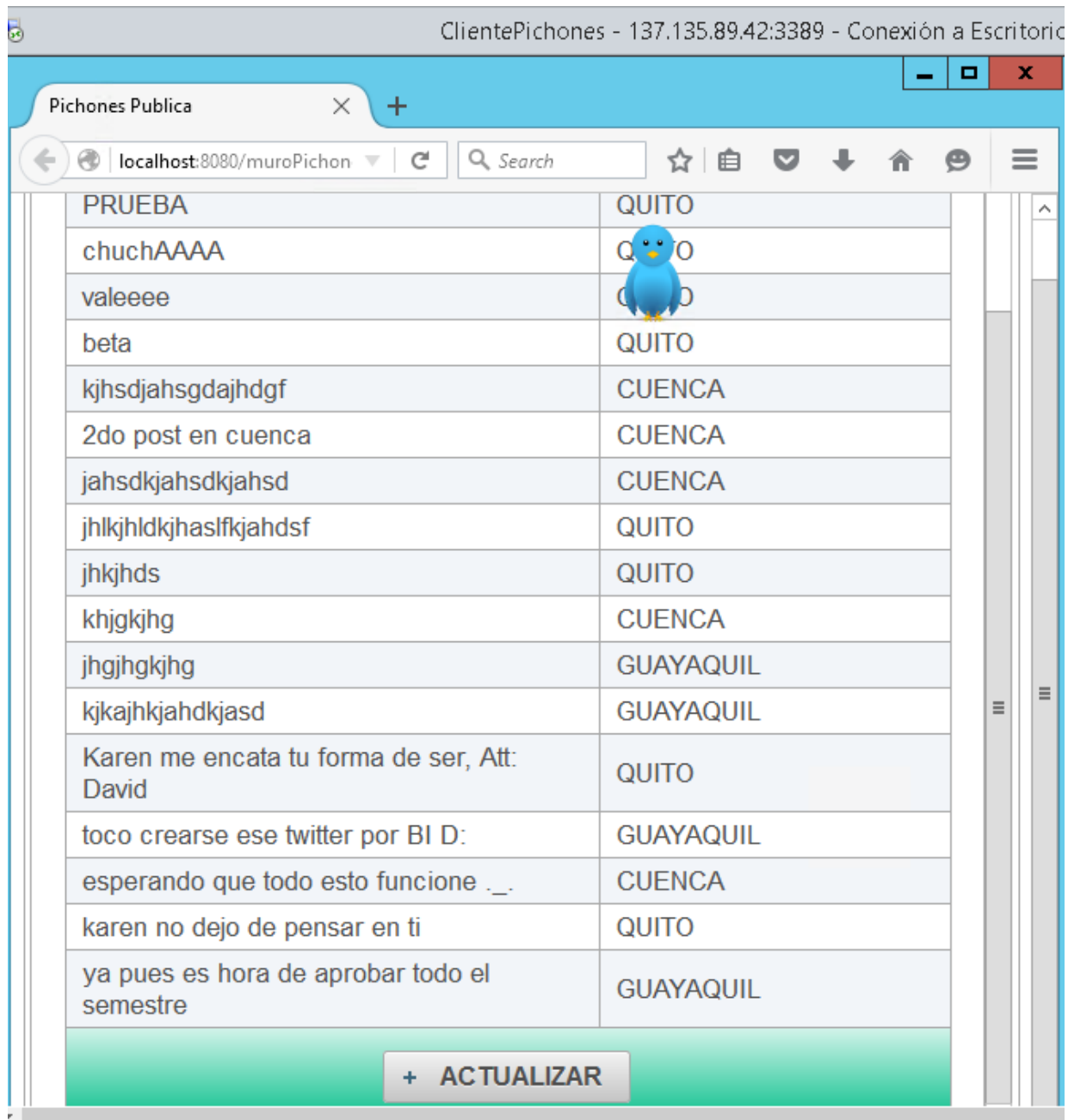
idPublicacion	lugGeoPublicacion	descripconPublicacion
12	GUAYAQUIL	jhgjhkgjhg
13	GUAYAQUIL	kjkajhkjahdkjasd
14	QUITO	Karen me encata tu forma ...
15	GUAYAQUIL	toco crearse ese twitter p...
16	CUENCA	esperando que todo esto f...
NULL	NULL	NULL

#### 4.6. Publicación Web

Finalmente la aplicación web muestra la publicación realizada seguida del lugar de su origen.

mandando directo a la base	QUITO
PRUEBA	QUITO
chuchAAAA	QUITO
valeeee	QUITO
beta	QUITO
kjhdsjahsgdajhdgf	CUENCA
2do post en cuenca	CUENCA
jahsdkjahsdkjahsd	CUENCA
jhlkjhlkdjhaslfdkjahdsf	QUITO
jhkjhds	QUITO
khjgkjhg	CUENCA
jhgjhkgkjhg	GUAYAQUIL
kjkajhkjahdkjasd	GUAYAQUIL
Karen me encata tu forma de ser, Att: David	QUITO
toco crearse ese twitter por BI D:	GUAYAQUIL
esperando que todo esto funcione ._.	CUENCA
+ ACTUALIZAR	

Las publicaciones son exitosas inclusive si es el servidor temporal el que está trabajando.



## 5. Conclusiones y trabajo futuro

- Existen algunas opciones para poder trabajar con la nube, se escogió Azure por la facilidad que nos ofrece al implementar las máquinas virtuales en la nube (IaaS) de forma amigable para nosotros como usuarios al momento de configurar la máquina virtual.
- Por nuestra arquitectura el servidor de respaldo nos ayuda con la disponibilidad e caso de que uno o más servidores no se encuentren en funcionamiento, pero en caso de que el servidor de respaldo igualmente no se encuentre funcionando, tenemos para futuro implementar que los otros servidores actúen también como respaldo, realizando también las configuraciones pertinentes en el balanceador para esta futura función y solventar una alta disponibilidad

- La aplicación actualmente se encuentra parametrizado de acuerdo a la ciudad de origen de las publicaciones pero se espera poder personalizar la clasificación acorde a las necesidades de la empresa o corporación que hagan uso de la misma.
- Se espera poder hacer del muro un lugar jerárquico en el cual se pueda desarrollar todo un desarrollo acorde a una publicación a tratar.
- Hacer una categorización de muro pichones según sea la importancia o quizás el tipo del tema a tratar.