

UNIVERSIDAD DE LAS AMÉRICAS - PUEBLA

Escuela de Ingeniería
Departamento de Ingeniería en Sistemas Computacionales

Presentación de la tesis titulada:

Creación de un gateway entre SMS y JMS para Kannel

Oscar Medina Duarte

Agenda

¿Qué?

- Introducción
- Definición del problema (Problemas en Kannel)
- Objetivo

¿Cómo?

- Solución (Reemplazar smsbox)
- Arquitectura
- Implementación

Después

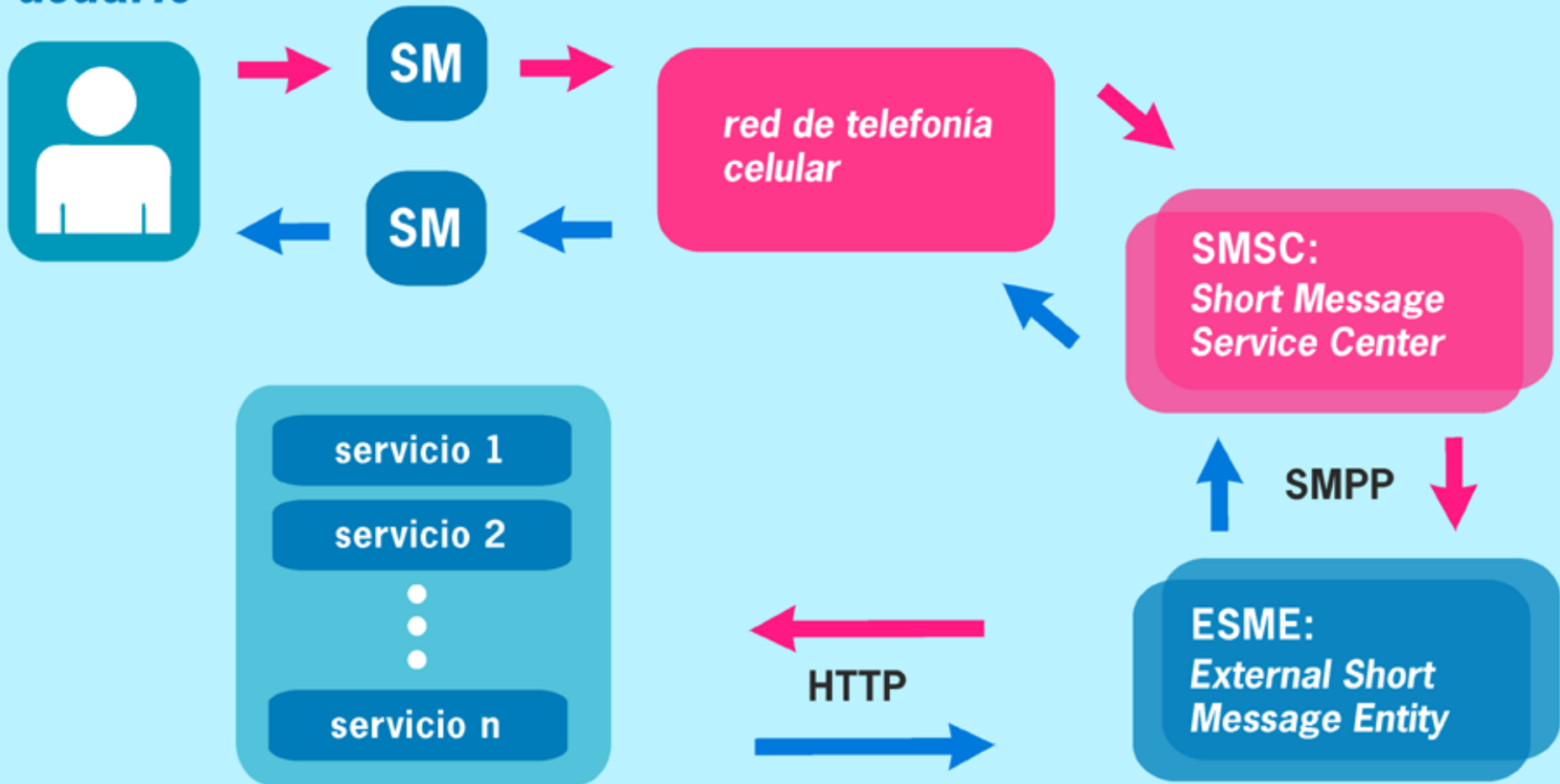
- Resultados
- Evaluación
- Aportación
- Trabajo a futuro

- SMS = Short Message Service
- JMS = Java Message Service
- Gateway
- Kannel

Dominio del problema

Introducción

usuario



Kannel

Componentes de Kannel

Introducción

SMPP



HTTP



Bearerbox



protocolo
interno



SMSbox

Problemas en Kannel

Definición
del
Problema

- Persistencia
- Transacciones
- Tamaño de los componentes
- Autonomía en los componentes

Protocolo interno de Kannel

Definición
del
Problema

- No documentado
- Basado en `struct`'s de C
- Basado en paquetes
- Los tipos de datos son variables concatenadas

Objetivo

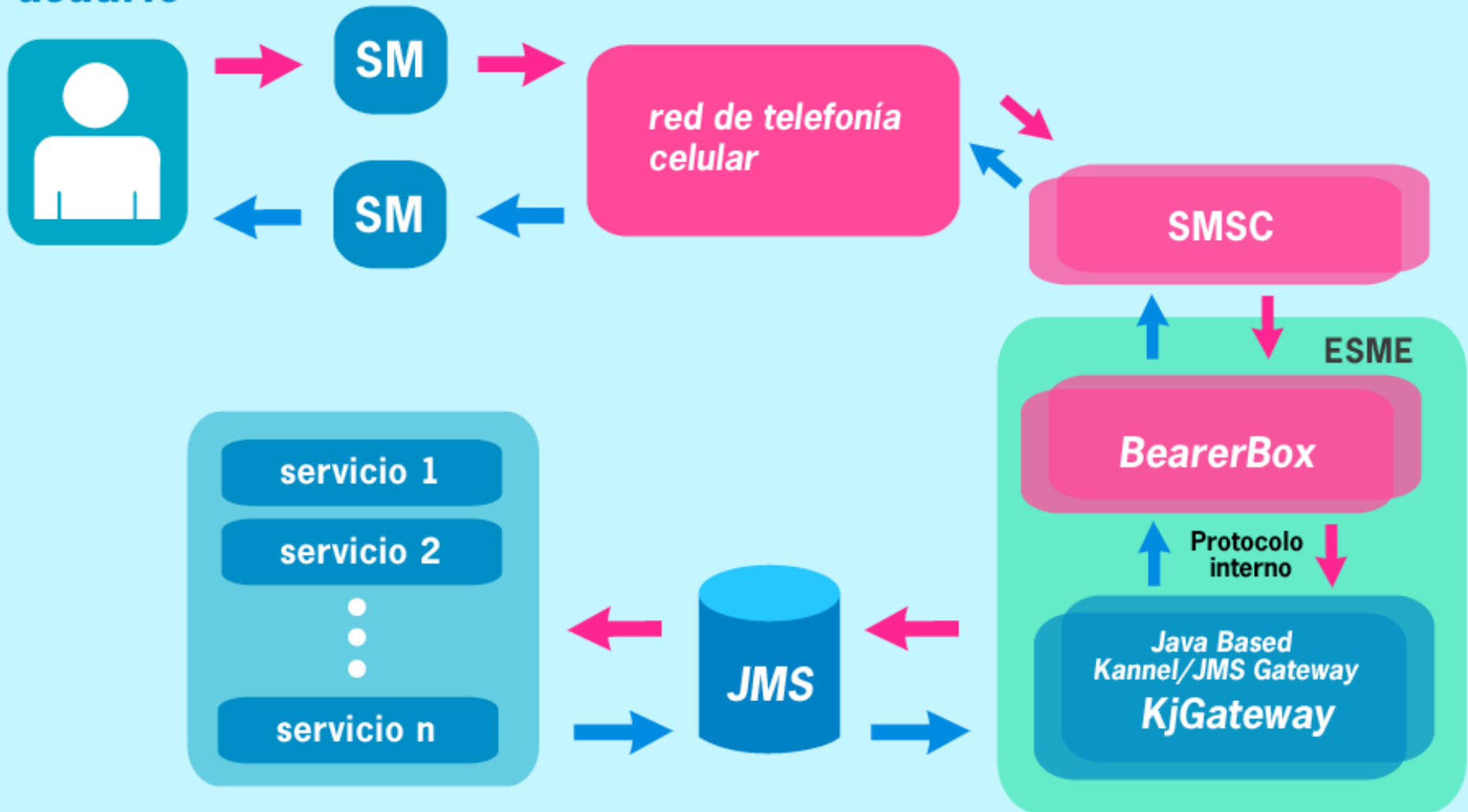
Objetivo

Crear un módulo para Kannel que sea un Gateway entre su protocolo interno y Java mediante el uso de JMS (Java Message Service), que permita persistencia de los mensajes transmitidos y uso de transacciones.

Reemplazar smsbox

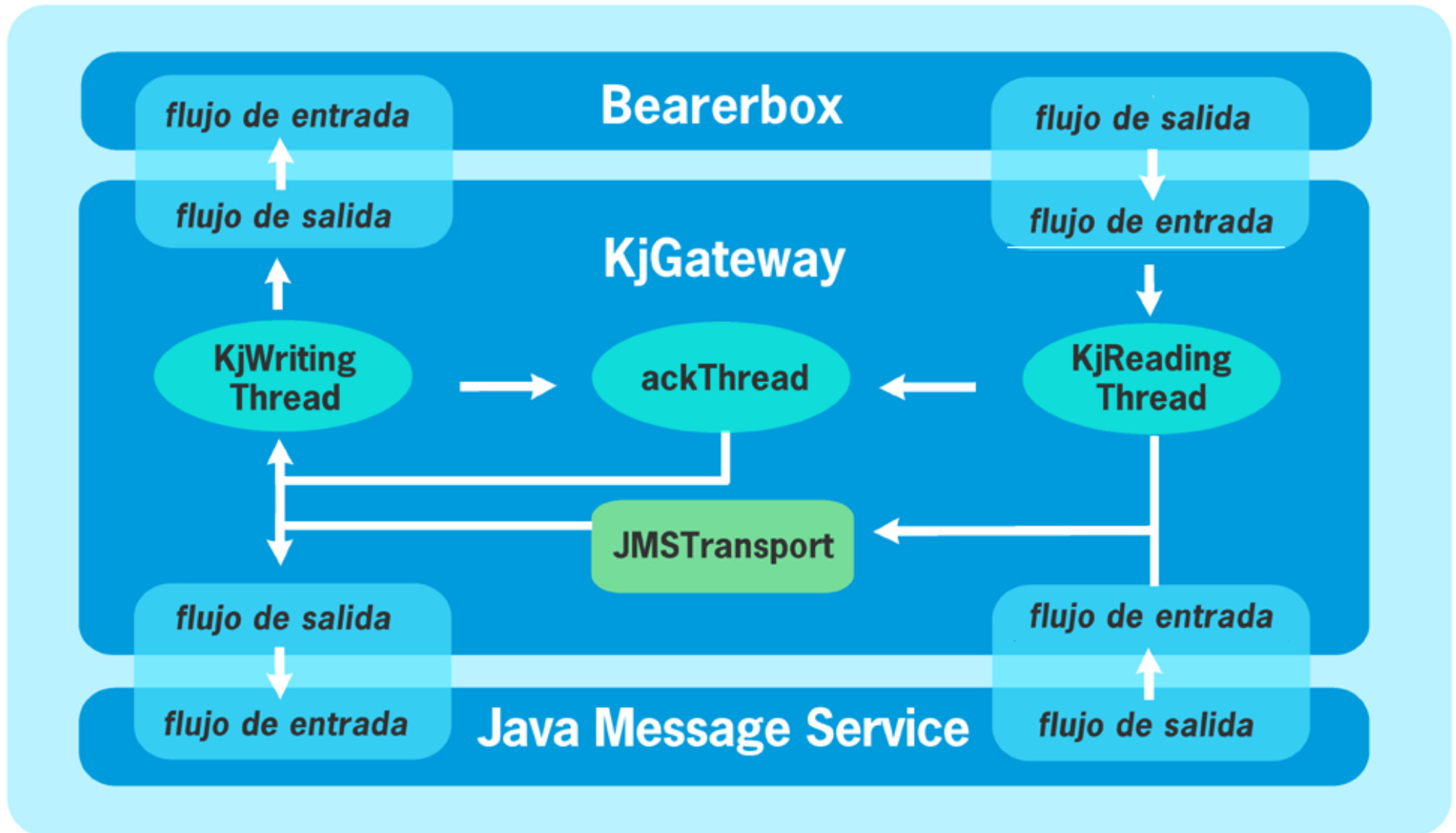
Solución

usuario



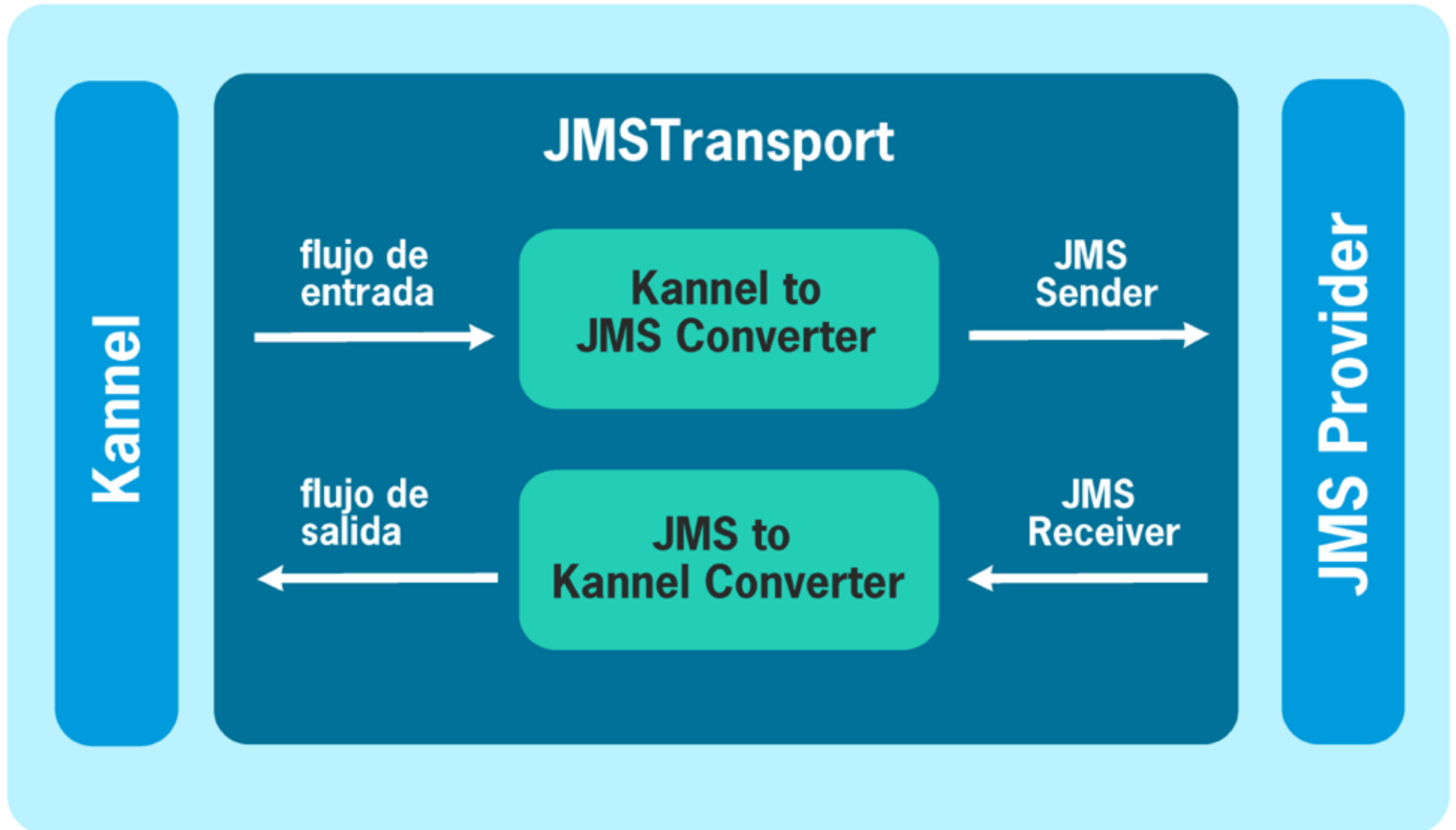
KjGateway

Arquitectura



JMSTransport

Arquitectura



KjProtocol API

Implementación

- Revisión del código fuente de Kannel
- Observación del tráfico generado por Kannel mediante el uso de un sniffer
- Documentación del protocolo (Apéndice A)
- Representación del protocolo en el paradigma orientado a Objetos

Paquetes

Implementación

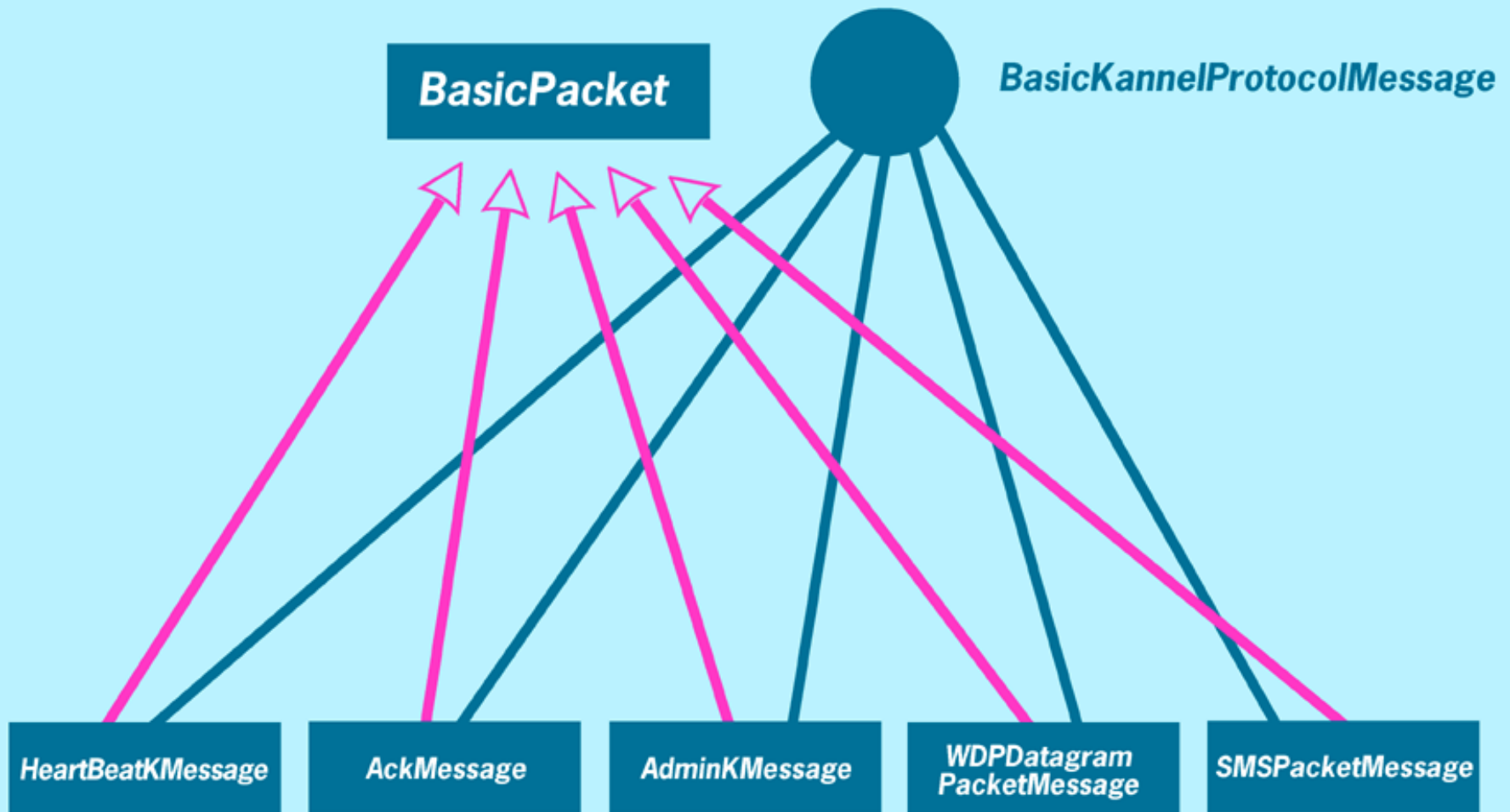
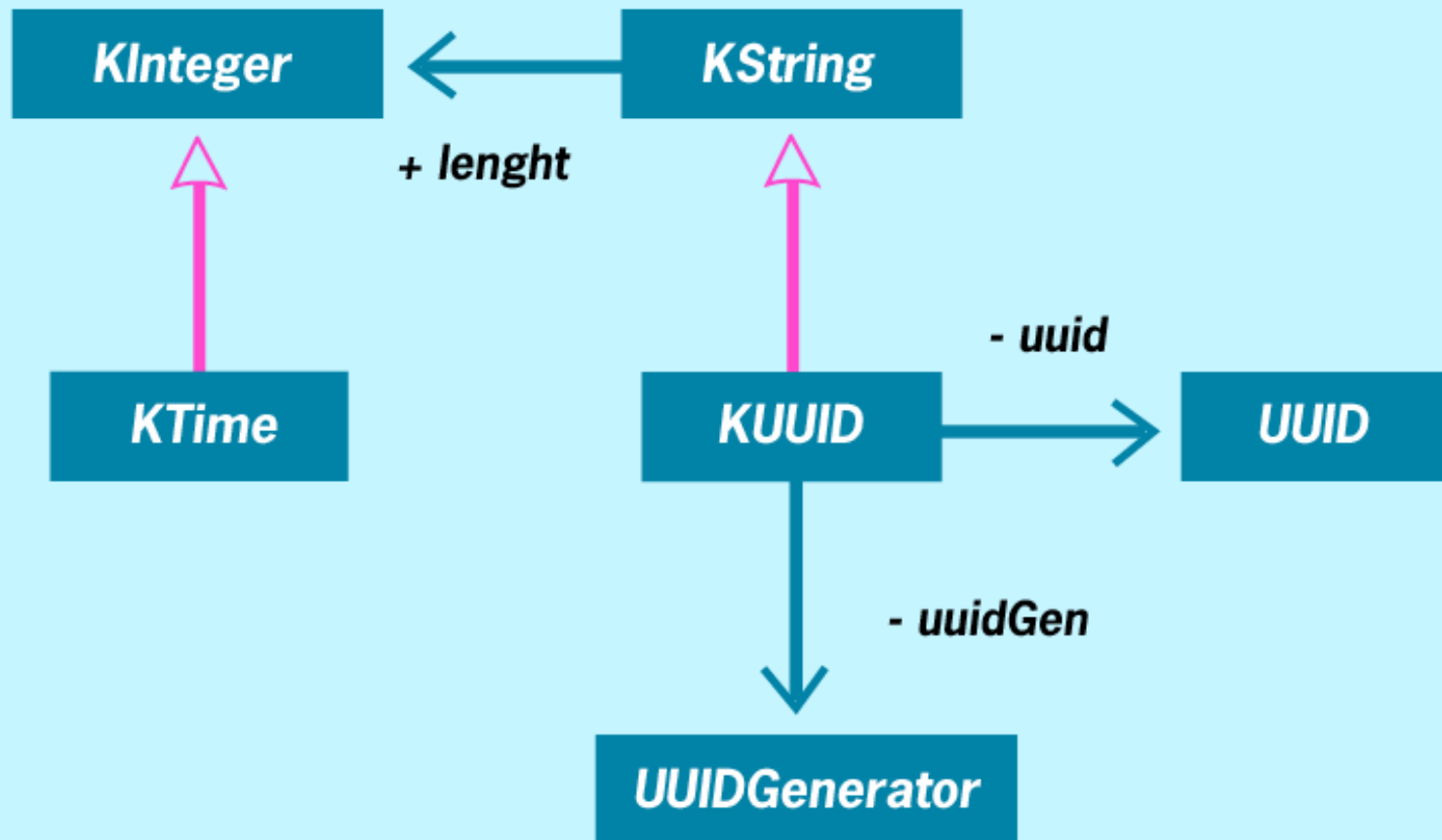


Diagrama de clases de la representación orientada a objetos de los paquetes

Tipos de datos

Implementación



Tipos de datos que definen a los paquetes

KannelBinding

Implementación

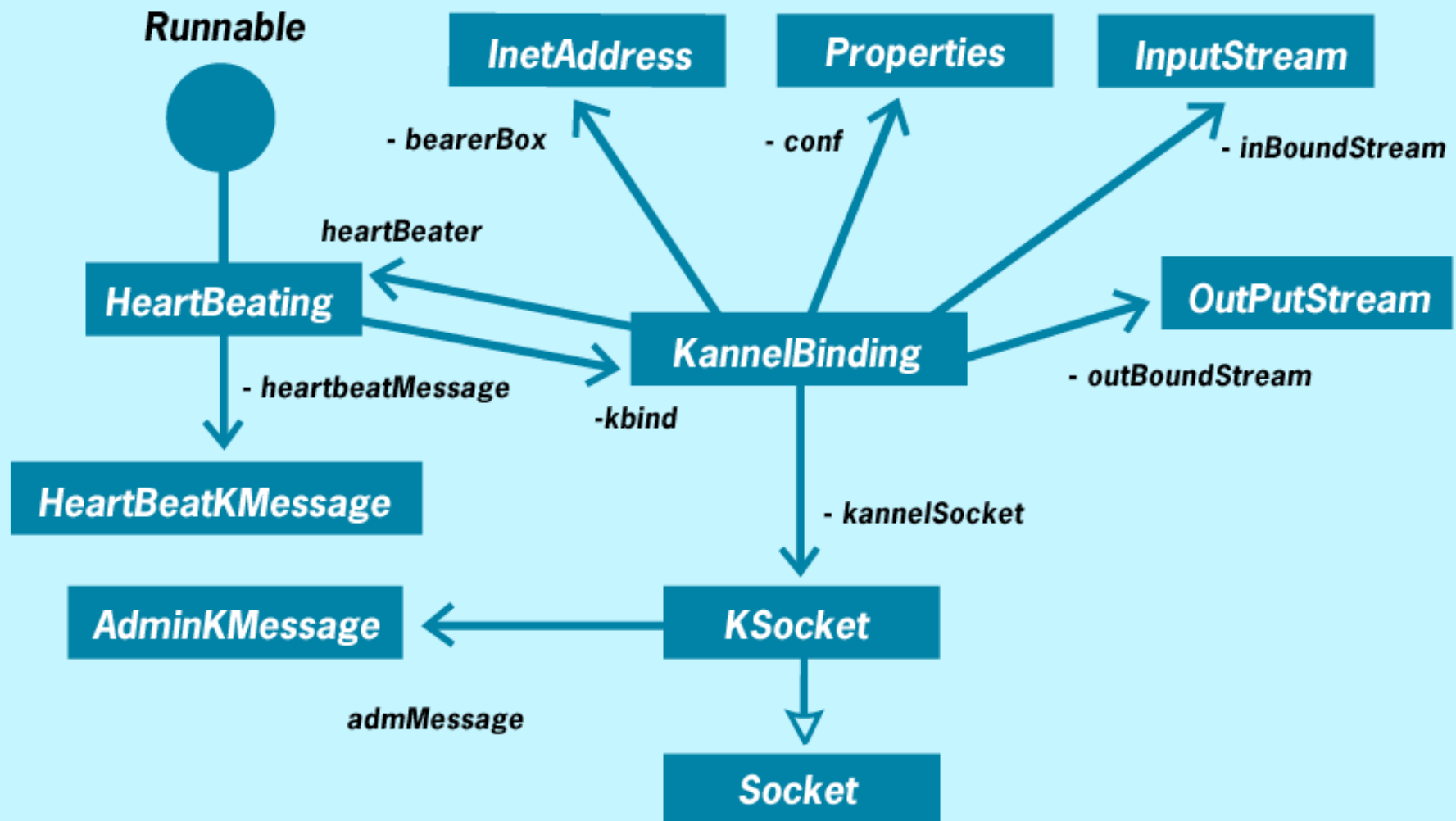


Diagrama de clases enfocado a KannelBinding

KannelBinding

Implementación

KannelBinding.readNext()

```
// kLen sera la longitud del paquete completo.
KInteger kLen = null;

// bInt es el arreglo que alojara los
// 4 bytes que formen dicho numero.
byte[] bInt = new byte[4];

int read = 0;
...
// Esta línea lee bInt.length bytes en bInt
// y regresa la cantidad total de bytes leídos
read = this.inBoundStream.read(bInt);
...
// Se parsean los 4 bytes en una clase
// que contiene dicha representación numérica
kLen = new KInteger(bInt);
len = kLen.getIntValue();
...
```

KannelBinding

Implementación

KannelBinding.readNext()

```
    ...  
    this.inBoundStream.read();  
    this.inBoundStream.read();  
    this.inBoundStream.read();  
    this.inBoundStream.read();  
    ...
```

KjGateway

Implementación

- Clase KannelJMSGateway
- Interfaz JMSTransport
- Clase KjReadingThread
- Clase KjWritingThread
- Clase AckCycleThread

KjReadingThread

Implementación

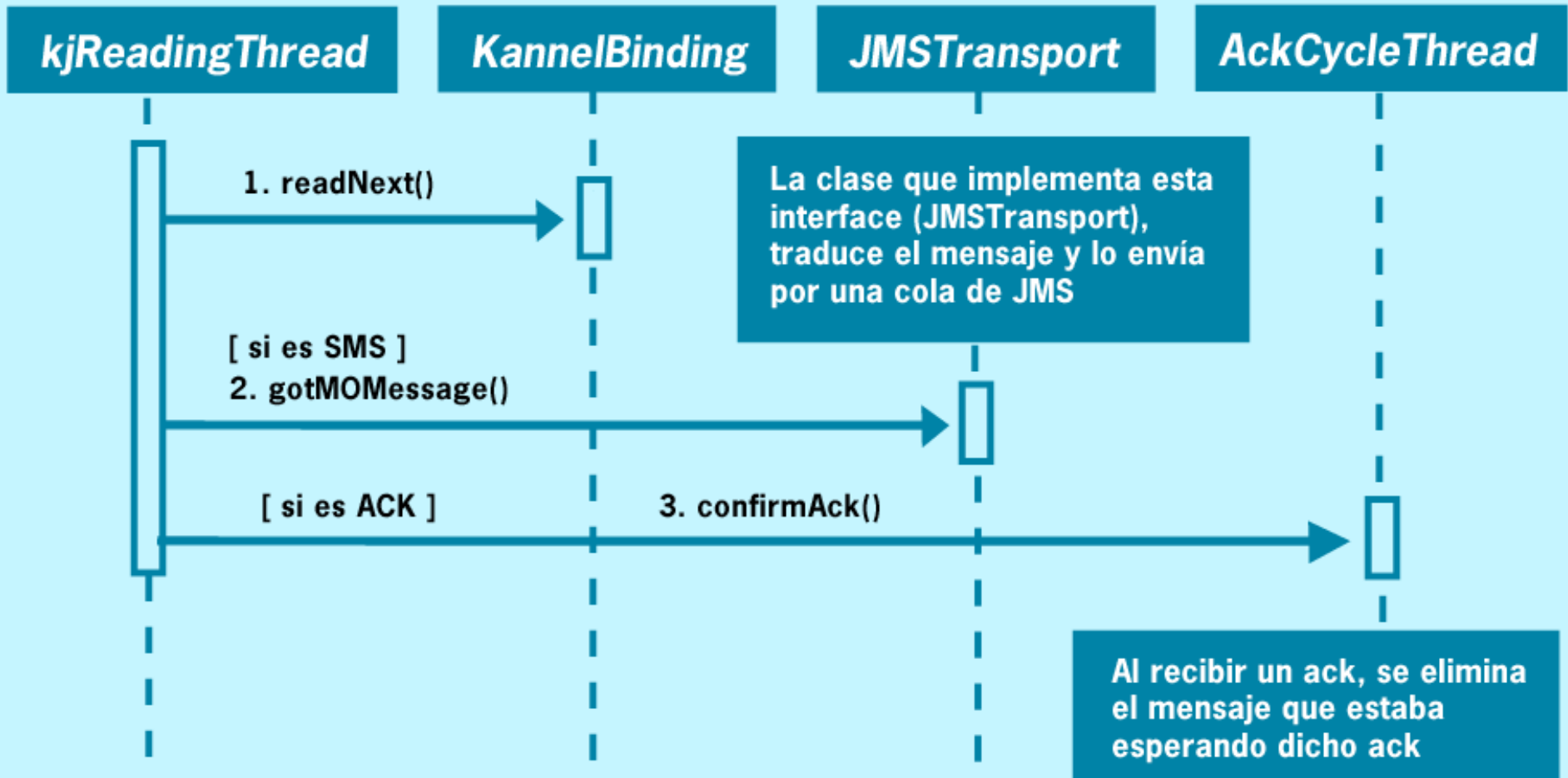


Diagrama de secuencias de KjReadingThread

KjWritingThread

Implementación

```
public void send(SMSPacketMessage pack){
    try{
        if(this.ackAdminThread != null){

            this.ackAdminThread.waitAck(pack);

        }

        this.kbind.writeNext((BasicKannelProtocolMessage)pack);

    }catch(IOException e){
        System.out.println("sendind message failed : " + e);
    }
}
```

AckCycleThread

Implementación

- Los mensajes que deben ser reenviados
- Los mensajes que han expirado
- Los mensajes que han sido recibidos

AckCycleThread

Implementación

Métodos:

- `public void confirmAck(KUUID uuid)`
- `public void waitAck(SMSPacketMessage smsPacketMessage)`

```
...
while(true){
    if (uuidList != null){
        ...
        currentTime = System.currentTimeMillis();
        for (; uuidList.hasMoreElements() ;){
            kuuid = uuidList.nextElement();

            // Compara tiempo de expiración del UUID
            // con el tiempo actual currentTime
            if( currentTime > ((Long)uuidHash.get(kuuid)).longValue() ){
                // Si expiro, lo removemos de la lista.
                uuidHash.remove(kuuid);
                smsHash.remove(kuuid);
                uuidList = uuidHash.keys();
            }else{
                outWriting.rawWrite(((SMSPacketMessage)smsHash.get(kuuid)).getMessage());
            }
        }
    }

    ...
    this.sleep(frecuency);
}catch(InterruptedException e){
    System.out.println("Ack waiting Interrupted...");
}
}
```

SimpleJMSTransport

Implementación

- Implementación de la interfaz JMSTransport
- Es donde se reciben los mensajes, se traducen y se re-direccionan.

SimpleService

Implementación

- Servicio de aplicación basado en JMS.
- Recibe un mensaje de texto, y si ese texto tiene una respuesta configurada, envía dicha respuesta al originador de la solicitud.
- Ejemplo simple que sirve como base para implementar servicios más complejos.

Pruebas realizadas

Pruebas

- Funcionamiento de la implementación del protocolo.
- Funcionamiento del Gateway
- Pruebas relacionadas

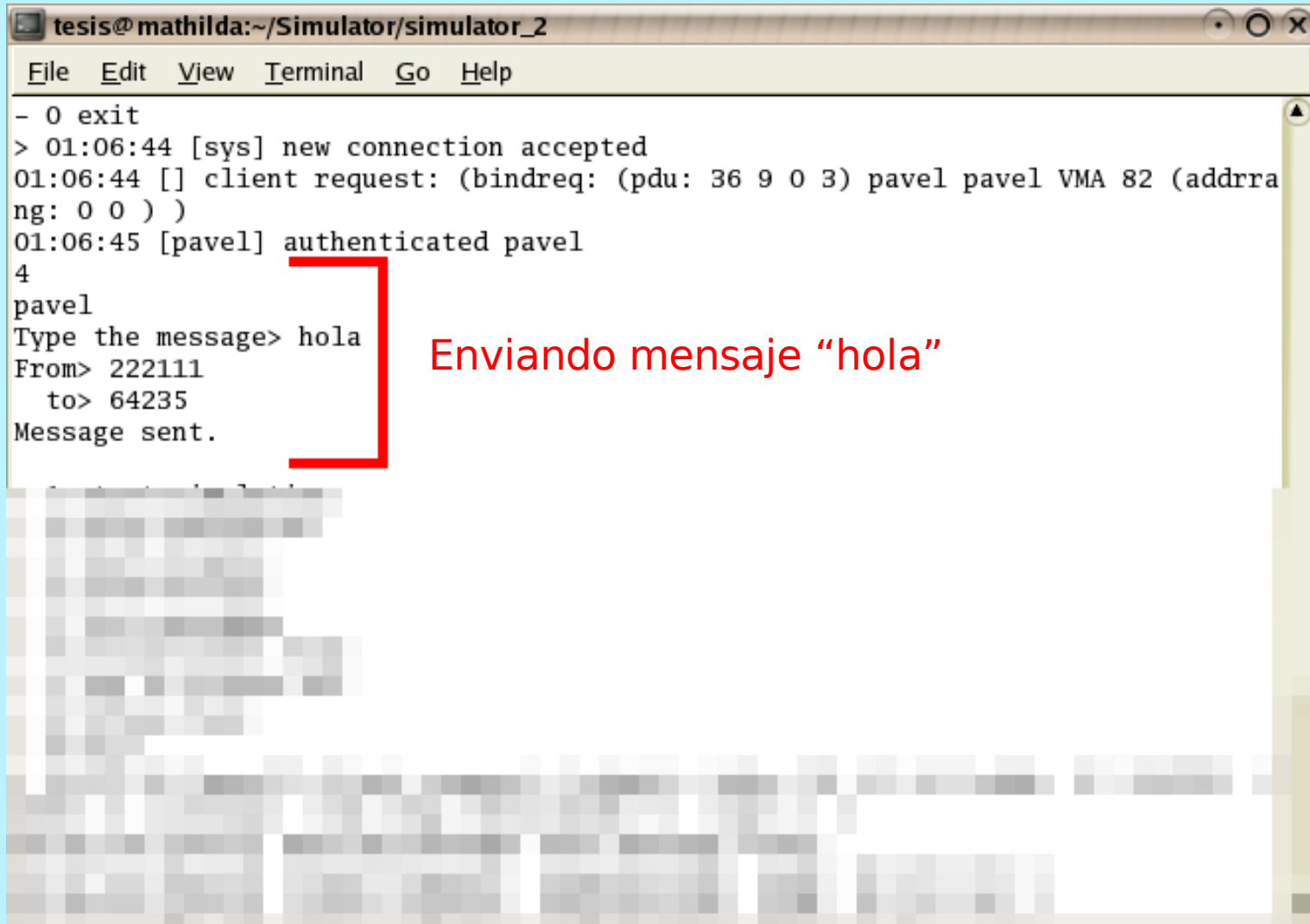
Ambiente de pruebas

Pruebas

1. Simulador de SMSC
2. bearerbox
3. JBossMQ
4. kjGateway
5. SimpleService

Enviando un mensaje

Pruebas



```
tesis@mathilda:~/Simulator/simulator_2
File Edit View Terminal Go Help
- 0 exit
> 01:06:44 [sys] new connection accepted
01:06:44 [] client request: (bindreq: (pdu: 36 9 0 3) pavel pavel VMA 82 (addr: 0 0 ))
01:06:45 [pavel] authenticated pavel
4
pavel
Type the message> hola
From> 222111
to> 64235
Message sent.
```

Enviando mensaje "hola"

Enviando un mensaje

Pruebas

```
Terminal
File Edit View Terminal Go Help
Terminal tesis@mathilda:~/jdkjGW/JBoss/jboss-4.0.1sp1/bin
2005-05-08 13:04:13 [4635] [12] DEBUG: Thread 12 (gw/bearerbox.c:wdp_router) maps to pid 4635.
2005-05-08 13:04:13 [4635] [12] DEBUG: udp_die: removing producers from udp-lists
2005-05-08 13:04:13 [4635] [12] DEBUG: Thread 12 (gw/bearerbox.c:wdp_router) terminates.
2005-05-08 13:04:13 [4635] [11] DEBUG: Thread 11 (gw/bb_udp.c:udp_sender) maps to pid 4635.
2005-05-08 13:06:54 [4635] [5] DEBUG: Started thread 13 (gw/bb_boxc.c:function)
2005-05-08 13:06:54 [4635] [13] DEBUG: Thread 13 (gw/bb_boxc.c:function) maps to pid 4635.
2005-05-08 13:06:54 [4635] [13] INFO: Client connected from <127.0.0.1>
2005-05-08 13:06:54 [4635] [13] DEBUG: Started thread 14 (gw/bb_boxc.c:boxc_sender)
2005-05-08 13:06:54 [4635] [14] DEBUG: Thread 14 (gw/bb_boxc.c:boxc_sender) maps to pid 4635.
2005-05-08 13:07:18 [4635] [14] DEBUG: send_msg: sending msg to box: <127.0.0.1>
2005-05-08 13:07:18 [4635] [14] DEBUG: boxc_sender: sent message to <127.0.0.1>
2005-05-08 13:07:19 [4635] [13] DEBUG: boxc_receiver: got ack
2005-05-08 13:07:19 [4635] [13] DEBUG: boxc_receiver: sms received
2005-05-08 13:07:19 [4635] [13] DEBUG: send_msg: sending msg to box: <127.0.0.1>
2005-05-08 13:07:24 [4635] [13] DEBUG: boxc_receiver: heartbeat with load value 0 received

```

Mensaje recibido en el bearerbox

Enviando un mensaje

Pruebas

```
tesis@mathilda:~/jkjGW/kjGateway
File Edit View Terminal Go Help
tesis@mathilda:~/jkjGW/kjGateway  tesis@mathilda:~/jkjGW/kjGateway
AckThread starting
Reading thread started.
out:
00000008 00000000 00000000 | .....

out:
00000008 00000000 00000000 | .....

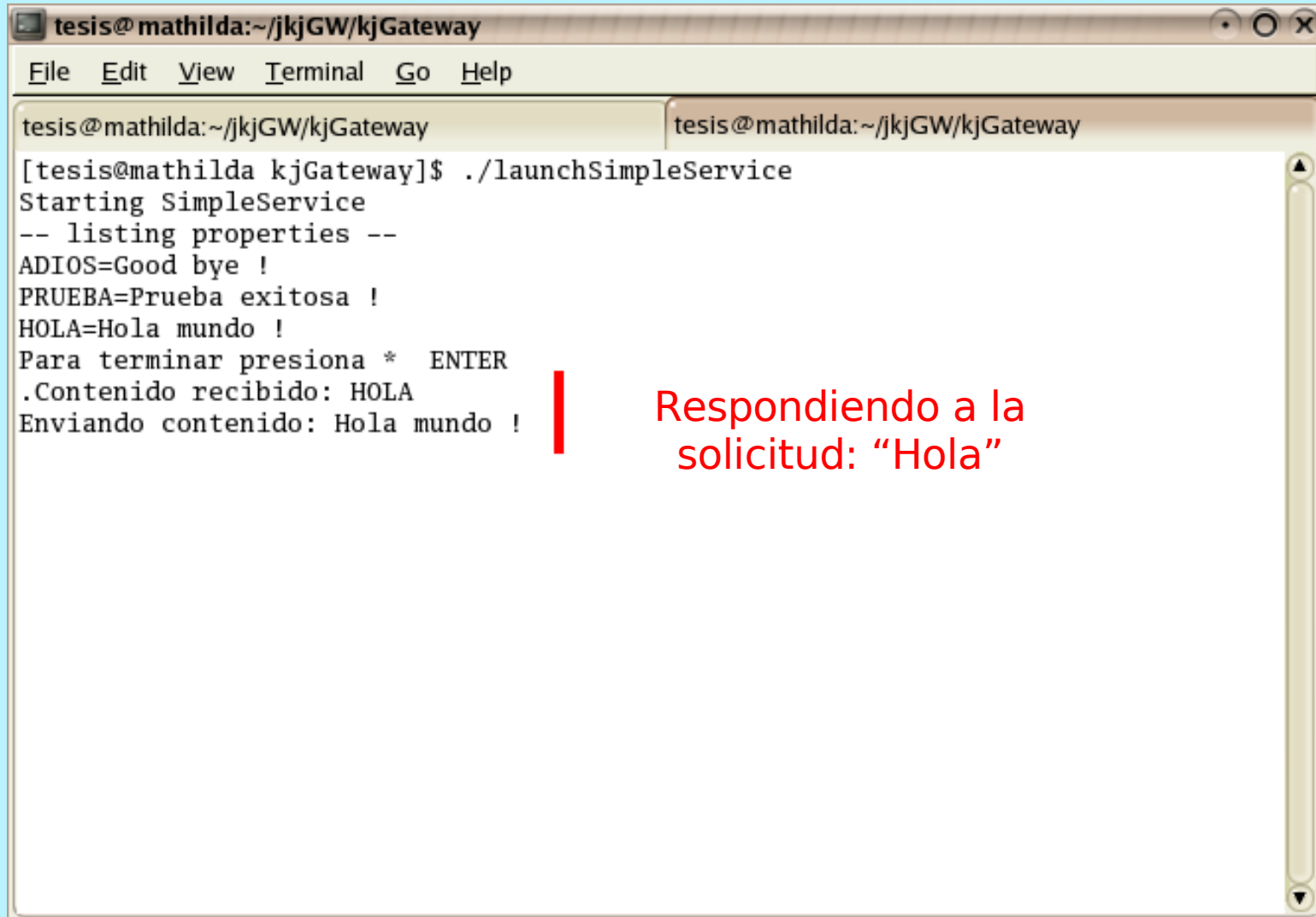
in:
000000A6 00000002 00000006 32323231 | ..... ..2221
31310000 00053634 323335FF FFFFFFF00 | 11....64 235....
00000468 6F6C6142 7E555600 00000774 | ...holaB .UV....t
6573745F 3030FFFF FFFFFFFF FFFF0000 | est_00.. .....
00246139 37323161 66312D35 3263372D | .$a9721a f1-52c7-
34653132 2D393931 312D6635 37346131 | 4e12-991 1-f574a1
39376266 33370000 0000FFFF FFFFFFFF | 97bf37.. .....
FFFF0000 00000000 0000FFFF FFFFFFFF | .....
FFFFFFFF FFFFFFFF FFFF0000 00000000 | .....
0000FFFF FFFFFFFF FFFFFFFF FFFFFFFF | .....
FFFFFFFF FFFF0000 0000 | .....

writing ack
ack written
.
```

Mensaje
recibido
por
kjGateway

Enviando un mensaje

Pruebas



```
tesis@mathilda:~/jkjGW/kjGateway
File Edit View Terminal Go Help
tesis@mathilda:~/jkjGW/kjGateway
[tesis@mathilda kjGateway]$ ./launchSimpleService
Starting SimpleService
-- listing properties --
ADIOS=Good bye !
PRUEBA=Prueba exitosa !
HOLA=Hola mundo !
Para terminar presiona * ENTER
.Contenido recibido: HOLA
Enviando contenido: Hola mundo !
```

Respondiendo a la solicitud: "Hola"

Respondiendo a un mensaje

Pruebas

```
tesis@mathilda:~/jkjGW/kjGateway
File Edit View Terminal Go Help
tesis@mathilda:~/jkjGW/kjGateway  tesis@mathilda:~/jkjGW/kjGateway

writing ack
ack written
.
out:
000000A7 00000002 00000005 36343233 | ..... ....6423
35000000 06323232 313131FF FFFFFFF00 | 5....222 111....
00000C48 6F6C6120 6D756E64 6F202142 | ...Hola. mundo.!B
7E5557FF FFFFFFFF FFFFFFFF FFFFFFF00 | .UW.....
00002430 34633361 6432342D 62666563 | ..$04c3a d24-bfec
2D313164 392D3961 35332D32 62623438 | -11d9-9a 53-2bb48
65633163 38653400 000001FF FFFFFFFF | ec1c8e4. ....
FFFFFFF0 00000000 000000FF FFFFFFFF | .....
FFFFFFF0 FFFFFFFF FFFFFFF0 00000000 | .....
000000FF FFFFFFFF FFFFFFFF FFFFFFFF | .....
FFFFFFF0 FFFFFFF0 000000 | .....

in:
00000034 00000003 00000000 427E5557 | ...4.... ....B.UW
00000024 30346333 61643234 2D626665 | ...$04c3 ad24-bfe
632D3131 64392D39 6135332D 32626234 | c-11d9-9 a53-2bb4
38656331 00000000 | 8ec1....

out:
```

Enviando
Mensaje de
respuesta
recibido por
el Servicio

Respondiendo a un mensaje

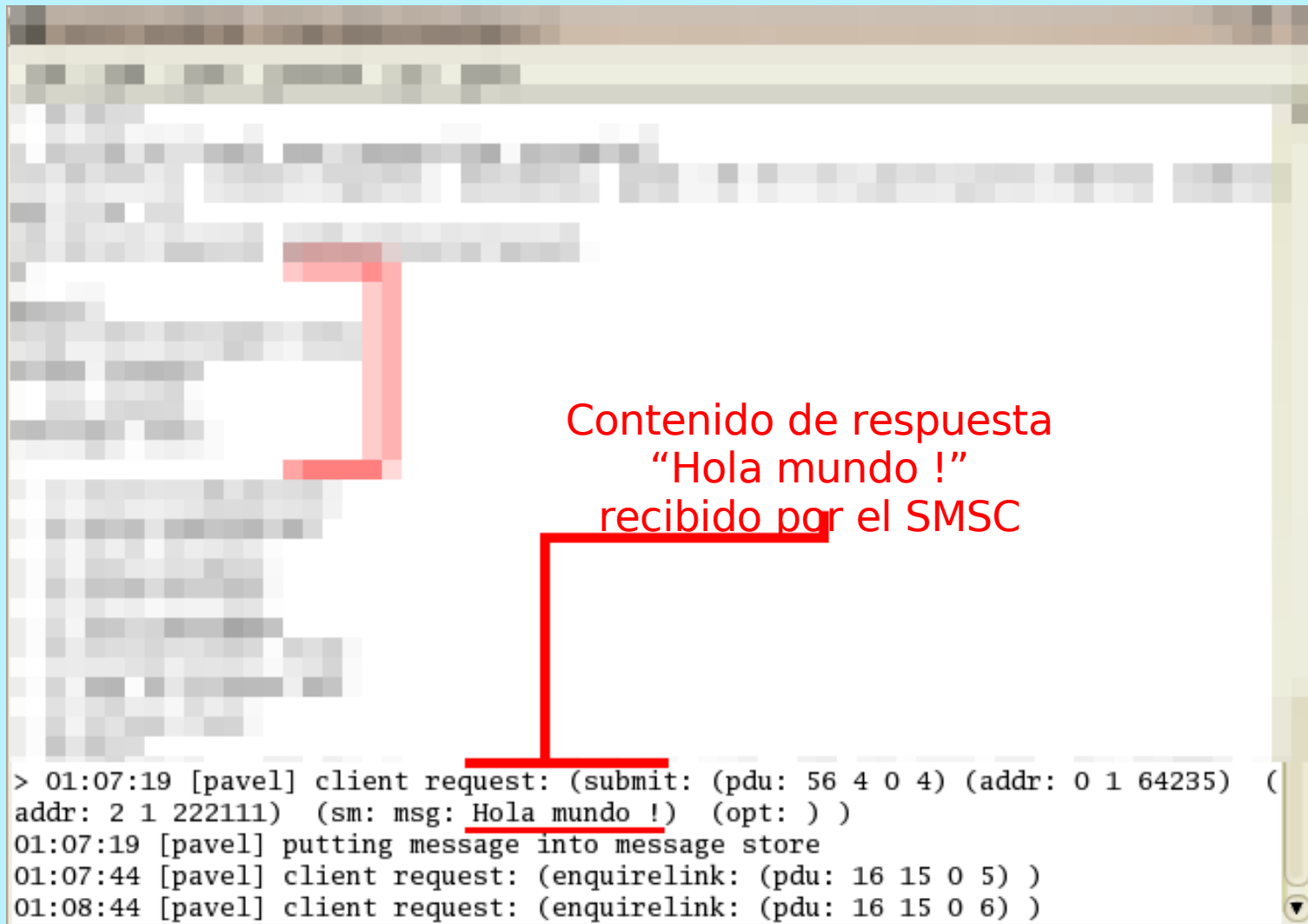
Pruebas

```
Terminal
File Edit View Terminal Go Help
Terminal tesis@mathilda:~/jkjGW/JBoss/jboss-4.0.1sp1/bin
2005-05-08 13:04:13 [4635] [12] DEBUG: Thread 12 (gw/bearerbox.c:wdp_router) maps to pid 4635.
2005-05-08 13:04:13 [4635] [12] DEBUG: udp_die: removing producers from udp-lists
2005-05-08 13:04:13 [4635] [12] DEBUG: Thread 12 (gw/bearerbox.c:wdp_router) terminates.
2005-05-08 13:04:13 [4635] [11] DEBUG: Thread 11 (gw/bb_udp.c:udp_sender) maps to pid 4635.
2005-05-08 13:06:54 [4635] [5] DEBUG: Started thread 13 (gw/bb_boxc.c:function)
2005-05-08 13:06:54 [4635] [13] DEBUG: Thread 13 (gw/bb_boxc.c:function) maps to pid 4635.
2005-05-08 13:06:54 [4635] [13] INFO: Client connected from <127.0.0.1>
2005-05-08 13:06:54 [4635] [13] DEBUG: Started thread 14 (gw/bb_boxc.c:boxc_sender)
2005-05-08 13:06:54 [4635] [14] DEBUG: Thread 14 (gw/bb_boxc.c:boxc_sender) maps to pid 4635.
2005-05-08 13:07:18 [4635] [14] DEBUG: send_msg: sending msg to box: <127.0.0.1>
2005-05-08 13:07:18 [4635] [14] DEBUG: boxc_sender: sent message to <127.0.0.1>
2005-05-08 13:07:19 [4635] [13] DEBUG: boxc_receiver: got ack
2005-05-08 13:07:19 [4635] [13] DEBUG: boxc_receiver: sms received
2005-05-08 13:07:19 [4635] [13] DEBUG: send_msg: sending msg to box: <127.0.0.1>
2005-05-08 13:07:24 [4635] [13] DEBUG: boxc_receiver: heartbeat with load value 0 received
```

Mensaje de respuesta recibido

Respondiendo a un mensaje

Pruebas



```
> 01:07:19 [pavel] client request: (submit: (pdu: 56 4 0 4) (addr: 0 1 64235) (
addr: 2 1 222111) (sm: msg: Hola mundo !) (opt: ) )
01:07:19 [pavel] putting message into message store
01:07:44 [pavel] client request: (enquirelink: (pdu: 16 15 0 5) )
01:08:44 [pavel] client request: (enquirelink: (pdu: 16 15 0 6) )
```

Contenido de respuesta
"Hola mundo !"
recibido por el SMSC

Resultados

- kjGateway
- kjProtocol API para crear aplicaciones usando el protocolo de Kannel bajo java
- Integración de servicios de mensajería celular usando JMS (Y todas sus ventajas)

Evaluación del Sistema

- HTTP Reemplazado por JMS
- Persistencia de Mensajes
- Posibilidad de uso de transacciones.
- Reducción del tamaño de los componentes.
- Aumento de la autonomía de los componentes.

Aportación

- Creación de un Gateway entre Kannel y JMS
- Simplificación del proceso de implementación de servicios SMS
- Documentación del protocolo interno de Kannel.
- Creación de un API para desarrollar módulos de Kannel en Java.

Trabajo a futuro

- Estudio de la aplicación y uso de transacciones en ambientes de cargo por acuse de recibido (MT billing).
- Exploración de arquitecturas para ambientes que requieren alta disponibilidad y tolerancia a fallas.
- Seguridad.
- Especificación formal de un protocolo para Kannel.



gracias !