

Anexo B. Configuraciones del Archivo extensions.conf

```
; Dialplan: PROTOTIPO DE UNA ESTACIÓN CELULAR PORTÁTIL PARA
; ATENCIÓN DE EMERGENCIAS
; Julián David Vásquez - Iván Fernando Santa
```

```
;Primer contexto. Verifica si un usuario ha marcado 1234 o alguna extensión
;distinta, si marca 1234 lo llevará a la línea de prioridad 1 con extensión
;start que se encuentra en el contexto dialplan, si marca otro número de 4
;cifras (patrón de marcado _XXXX) lo llevará al macro-dialSIP, con dos
;argumentos, la cadena "NR" concatenada con el número marcado, y el número
;marcado.
```

[sip-local]

```
exten => 1234,1,Goto(dialplan,start,1)
exten => _XXXX,1,Macro(dialSIP,${NR}${EXTEN},${EXTEN})
```

```
;Luego de que el usuario digita 1234 para registrarse entonces Asterisk
;validará si ya hay IMSI guardado en cada uno de las extensiones disponibles
;para así asignar una de estas al usuario que solicita.
;Así se irán asignando extensiones en orden hasta que ya no hayan más, momento
;en el cual se escuchará un mensaje indicando que no hay más extensiones
;disponibles. Por defecto asignarán un máximo de 12 extensiones.
;Al verificar la variable se salta a una extensión con el número de la
;extensión que se va a asignar pero con un cero adelante, para facilidad de
;comprensión del código.
```

[dialplan]

```
exten => start,1,GotoIf($[[${LEN(${NR1001})} = 0]]?01001,1)
    same => n,GotoIf($[[${LEN(${NR1002})} = 0]]?01002,1)
    same => n,GotoIf($[[${LEN(${NR1003})} = 0]]?01003,1)
    same => n,GotoIf($[[${LEN(${NR1004})} = 0]]?01004,1)
    same => n,GotoIf($[[${LEN(${NR1005})} = 0]]?01005,1)
    same => n,GotoIf($[[${LEN(${NR1006})} = 0]]?01006,1)
    same => n,GotoIf($[[${LEN(${NR1007})} = 0]]?01007,1)
    same => n,GotoIf($[[${LEN(${NR1008})} = 0]]?01008,1)
    same => n,GotoIf($[[${LEN(${NR1009})} = 0]]?01009,1)
    same => n,GotoIf($[[${LEN(${NR1010})} = 0]]?01010,1)
    same => n,GotoIf($[[${LEN(${NR1011})} = 0]]?01011,1)
    same => n,GotoIf($[[${LEN(${NR1012})} = 0]]?01012,1)
    same => n,NoOp(No hay numeros disponibles en el momento)
    same => n,Playback(noNumbers)
    same => n,Hangup()
```

```
;En este bloque se asigna el IMSI del teléfono a la variable global que la
;almacenará.
;La variable NR1001 corresponde a la extensión 1001 y así sucesivamente.
;El CALLERID(num) entrega el IMSI del teléfono que está intentando
;registrarse.
;Luego de almacenar la variable, Asterisk cuelga.
```

```
exten => 01001,1,Set(GLOBAL(NR1001)=${CALLERID(num)})
    same => n,Hangup()

exten => 01002,1,Set(GLOBAL(NR1002)=${CALLERID(num)})
    same => n,Hangup()

exten => 01003,1,Set(GLOBAL(NR1003)=${CALLERID(num)})
    same => n,Hangup()

exten => 01004,1,Set(GLOBAL(NR1004)=${CALLERID(num)})
    same => n,Hangup()

exten => 01005,1,Set(GLOBAL(NR1005)=${CALLERID(num)})
    same => n,Hangup()
```

```

exten => 01006,1,Set(GLOBAL(NR1006)=${CALLERID(num)})
      same => n,Hangup()

exten => 01007,1,Set(GLOBAL(NR1007)=${CALLERID(num)})
      same => n,Hangup()

exten => 01008,1,Set(GLOBAL(NR1008)=${CALLERID(num)})
      same => n,Hangup()

exten => 01009,1,Set(GLOBAL(NR1009)=${CALLERID(num)})
      same => n,Hangup()

exten => 01010,1,Set(GLOBAL(NR1010)=${CALLERID(num)})
      same => n,Hangup()

exten => 01011,1,Set(GLOBAL(NR1011)=${CALLERID(num)})
      same => n,Hangup()

exten => 01012,1,Set(GLOBAL(NR1012)=${CALLERID(num)})
      same => n,Hangup()

```

;Ahora que las extensiones han sido registradas, vamos al comportamiento ;cuando un usuario intenta marcar a otra extensión. Primero se verifica que ;la persona que está intentando realizar la llamada efectivamente esté ;autenticada al sistema, es decir, ha recibido una extensión. ;Al realizar la comparación, si está concuerda con alguna de las variables NR, ;se irá a la extensión etiquetada con el nombre de dicha variable, si no ;concuerda con ninguna, lo enviará a la etiqueta de no registrado donde le ;leerá un mensaje que indica que se debe marcar primero al 1234.

[macro-dialSIP]

```

exten => s,1,GotoIf($[[${NR1001} = ${CALLERID(num)}]]?NR1001)
      same => n,GotoIf($[[${NR1002} = ${CALLERID(num)}]]?NR1002)
      same => n,GotoIf($[[${NR1003} = ${CALLERID(num)}]]?NR1003)
      same => n,GotoIf($[[${NR1004} = ${CALLERID(num)}]]?NR1004)
      same => n,GotoIf($[[${NR1005} = ${CALLERID(num)}]]?NR1005)
      same => n,GotoIf($[[${NR1006} = ${CALLERID(num)}]]?NR1006)
      same => n,GotoIf($[[${NR1007} = ${CALLERID(num)}]]?NR1007)
      same => n,GotoIf($[[${NR1008} = ${CALLERID(num)}]]?NR1008)
      same => n,GotoIf($[[${NR1009} = ${CALLERID(num)}]]?NR1009)
      same => n,GotoIf($[[${NR1010} = ${CALLERID(num)}]]?NR1010)
      same => n,GotoIf($[[${NR1011} = ${CALLERID(num)}]]?NR1011)
      same => n,GotoIf($[[${NR1012} = ${CALLERID(num)}]]?NR1012:NOT_LOGGED_IN)

```

;En esta etapa se reemplaza el CALLERID(num) y CALLERID(name) por el número de ;extensión, para evitar que salga el IMSI en la pantalla del usuario llamado. ;Luego de esto va a la etiqueta CONTINUE

```

exten => s,n(NR1001),Set(CALLERID(num)=1001)
      same => n,Set(CALLERID(name)=1001)
      same => n,Goto(CONTINUE)

exten => s,n(NR1002),Set(CALLERID(num)=1002)
      same => n,Set(CALLERID(name)=1002)
      same => n,Goto(CONTINUE)

exten => s,n(NR1003),Set(CALLERID(num)=1003)
      same => n,Set(CALLERID(name)=1003)
      same => n,Goto(CONTINUE)

exten => s,n(NR1004),Set(CALLERID(num)=1004)
      same => n,Set(CALLERID(name)=1004)
      same => n,Goto(CONTINUE)

exten => s,n(NR1005),Set(CALLERID(num)=1005)

```

```

same => n,Set(CALLERID(name)=1005)
same => n,Goto(CONTINUE)

exten => s,n(NR1006),Set(CALLERID(num)=1006)
same => n,Set(CALLERID(name)=1006)
same => n,Goto(CONTINUE)

exten => s,n(NR1007),Set(CALLERID(num)=1007)
same => n,Set(CALLERID(name)=1007)
same => n,Goto(CONTINUE)

exten => s,n(NR1008),Set(CALLERID(num)=1008)
same => n,Set(CALLERID(name)=1008)
same => n,Goto(CONTINUE)

exten => s,n(NR1009),Set(CALLERID(num)=1009)
same => n,Set(CALLERID(name)=1009)
same => n,Goto(CONTINUE)

exten => s,n(NR1010),Set(CALLERID(num)=1010)
same => n,Set(CALLERID(name)=1010)
same => n,Goto(CONTINUE)

exten => s,n(NR1011),Set(CALLERID(num)=1011)
same => n,Set(CALLERID(name)=1011)
same => n,Goto(CONTINUE)

exten => s,n(NR1012),Set(CALLERID(num)=1012)
same => n,Set(CALLERID(name)=1012)
same => n,Goto(CONTINUE)

;Acá llegaran los intentos de llamadas de usuarios que no se han registrado
;con el 1234.

exten => s,n(NOT_LOGGED_IN),NoOp(El teléfono del que está llamando no se ha
logueado al sistema)
same => n,Set(TIMEOUT(absolute)=15)
same => n,PlayBack(silence/2)
same => n,MusicOnHold()
same => n,Hangup()

;En este tramo del dialplan se define la variable STOP en 0, luego se verifica
;el numero al que se está marcando (ARG1 es la concatenación de "NR" y el
;número de extensión que ha marcado el usuario,
;es decir el nombre de la variable de extensión a la cual se está marcando.
;Si dicha variable está vacía, se irá a la etiqueta STOP, de lo contrario
;seguirá con el dialplan.

exten => s,n(CONTINUE),Set(STOP=0)
same => n,NoOp(${LEN(${ARG1}})
same => n,GotoIf(${[${LEN(${ARG1}}] = ${STOP}}]?STOPCALL)

;Así, el flujo de llamada continua, estableciéndose la comunicación entre los
;dos usuarios con un tiempo límite de 5 minutos (300 segundos), un timeout de
;espera de contestado de 20 y las opciones rt.
;En pantalla se imprimirá el estado de la llamada (Variable DIALSTATUS), al
;pasar los 20 segundo se irá a la etiqueta "s-" concatenada con la variable
;DIALSTATUS.

exten => s,n,NoOp(El flujo de llamada continua)
same => n,Set(TIMEOUT(absolute)=300)
same => n,Dial(SIP/${ARG1},20,rt)
same => n,NoOp(El estado es ${DIALSTATUS})
same => n,Goto(s-${DIALSTATUS},1)

;Etiqueta STOPCALL: acá llegaran las llamadas si al número que se está
;llamando no se encuentra registrado.

```

```
exten => s,n(STOPCALL),NoOp(El número al cual se esta marcando no está registrado)
    same => n,Playback(silence/2)
    same => n,Set(TIMEOUT(absolute)=15)
    same => n,Answer()
    same => n,MusicOnHold()
    same => n,Hangup()
```

;Acá llegarán las llamadas en el caso en que al pasar los 20 segundo no haya
;respuesta por alguna razón.
;Depende de lo que arroje la variable DIALSTATUS.

```
exten => s-CANCEL,1,Hangup()
```

```
exten => s-NOANSWER,1,Hangup()
```

```
exten => s-BUSY,1,Busy(30)
    same => n,Hangup()
```

;En el caso en que el DIALSTATUS haya retornado congestión, se esperarán 10
;segundos, luego de esto, si persiste, se enviará la llamada a canal no
;disponible CHANUNAVAIL.

```
exten => s-CONGESTION,1,Congestion(10)
    same => n,NoOp(la extensión esta congestionada)
    same => n,Goto(s-CHANUNAVAIL,1)
    same => n,Hangup()
```

;A este destino llegarán las llamadas si el DIALSTATUS entrega el valor
;CHANUNAVAIL, o si luego de que DIALSTATUS haya entregado CONGESTION y se
;hayan esperado 10 segundos.
;Lo que se hará es entregar un mensaje al usuario y luego de esto se liberará
;la variable ;NR que contiene el IMSI del teléfono que se ya no se encuentra
;disponible.

```
exten => s-CHANUNAVAIL,1,NoOp(EL teléfono esta registrado pero lo han apagado  
o esta fuera de rango)
    same => n,Playback(silence/2)
    same => n,Playback(ss-noservice)
    same => n,Goto(s-${ARG2},1)
```

;En este bloque se restauran las variables luego de ser marcadas como no
;disponibles.

```
exten => s-1001,1,Set(GLOBAL(NR1001)=)
    same => n,Hangup()
```

```
exten => s-1002,1,Set(GLOBAL(NR1002)=)
    same => n,Hangup()
```

```
exten => s-1003,1,Set(GLOBAL(NR1003)=)
    same => n,Hangup()
```

```
exten => s-1004,1,Set(GLOBAL(NR1004)=)
    same => n,Hangup()
```

```
exten => s-1005,1,Set(GLOBAL(NR1005)=)
    same => n,Hangup()
```

```
exten => s-1006,1,Set(GLOBAL(NR1006)=)
    same => n,Hangup()
```

```
exten => s-1007,1,Set(GLOBAL(NR1007)=)
    same => n,Hangup()
```

```
exten => s-1008,1,Set(GLOBAL(NR1008)=)
    same => n,Hangup()
```

```
exten => s-1009,1,Set(GLOBAL(NR1009=))
      same => n,Hangup()

exten => s-1010,1,Set(GLOBAL(NR1010=))
      same => n,Hangup()

exten => s-1011,1,Set(GLOBAL(NR1011=))
      same => n,Hangup()

exten => s-1012,1,Set(GLOBAL(NR1012=))
      same => n,Hangup()
```

Anexo A. Configuraciones del archivo sip.conf

[general]

```
language = es           ;Lenguaje en el que se reproducirán los
mensajes prediseñados
bindport = 5060         ;Puerto UDP por donde se realizarán los
conexiones SIP
bindaddr = 0.0.0.0      ;Direcciones IP desde donde se recibirán
conexiones (cualquiera)
autocreatepeer = yes    ;Permite autocreación de usuarios
canreinvite = no       ;Fuerza a Asterisk a administrar todo el flujo de
datos
nat = yes              ;Determina si el dispositivo está detrás de un
NAT
call-limit = 1         ;Define la cantidad de llamadas que se tendrán
en el canal
type = friend          ;Tipo de usuario, define el comportamiento de
los usuarios o canales
allow = gsm            ;Define el codec gsm por defecto para la
codificación de audio
context = sip-local    ;Contexto al cual se llevarán las llamadas
host = 127.0.0.1       ;Se asume que los servicios están en la misma
máquina
```

[canal-global] (!)

```
type = friend
context = sip-local
host = dynamic         ;Indica que el host puede cambiar su
direccionamiento y parámetros
canreinvite = no
disallow = all         ;Deshabilita todos los codecs de audio
allow = gsm            ;Habilita el codec gsm
dtmfmode = rfc2833    ;Modo de reconocimiento de tonos dtmf
```

[IMS1732101018239328] (canal-global)