UNIVERSIDADE
DE VIGO

ESCOLA SUPERIOR DE ENXEÑARÍA INFORMÁTICA

Memoria do Traballo de Fin de Grao que presenta

D. Álvaro López Borrajo

para a obtención do Título de Graduado en Enxeñaría Informática

Túnel VPN restrinxido a servizos determinados

Xuño, 2019

Traballo de Fin de Grao Nº: EI 18/19-012

Titor/a: Miguel Ramón Díaz-Cacho Medina
Área de coñecemento: Enxeñería de Sistemas e Automática
Departamento: Enxeñería de Sistemas e Automática

2019

Dedicado a tódala comunidade de Linux. Este é o meu primeiro regalo de volta de moitos.

Moitas gracias, Paula, por estar ó meu carón sempre que o precisei durante todo este tempo.

Índice

Índice de ilustracións

1. Introdución

A información é un dereito humano, e grazas a Internet, está ao alcance de cada vez máis persoas no mundo.

Porén, en moitos países, como por exemplo China, Rusia, Turquía, Arabia Saudita, ou, máis notablemente, Corea do Norte, o acceso a Internet está restrinxido e censurado para que os gobernos destes países poidan controlar as comunicacións.

Habitualmente, empréganse conexións VPN para evitar esta censura.

As VPN (Virtual Private Network) son redes privadas extendidas a través dunha rede pública (como pode ser Internet), e permiten aos usuarios acceder á rede privada dende a rede pública coma se estivesen conectados directamente á rede privada.

Isto lógrase tunelando ou encapsulando o tráfico dirixido cara a rede privada e transmitíndoo cara o servidor VPN, que desencapsulará o tráfico e o difundirá na rede privada.

Esta tecnoloxía foi desenvolta orixinalmente para permitir que usuarios remotos ou redes de sucursales dunha empresa poidan acceder aos recursos e aplicacións dunha rede privada empresarial coma se estivesen presentes físicamente.

Sen embargo, as VPN popularizáronse debido a que o tráfico habitualmente está cifrado, ademais de encapsulado, permitindo ocultar os seus contidos e o destino.

Pero presentan un problema: Todo o tráfico do equipo envíase a través do túnel.

Non é nada habitual que un equipo manteña unha única conexión a un só porto dun único servidor durante períodos prolongados de tempo, por isto, as autoridades son capaces de detectar se un equipo está conectado a un VPN, polo que o uso típico das VPN resulta inefectivo se se pretende evitar a censura.

A proposta deste TFG da unha solución a este problema, permitindo manter simultáneamente tráfico enrutado cara a unha VPN e tráfico sen enrutar.

2. Obxectivos

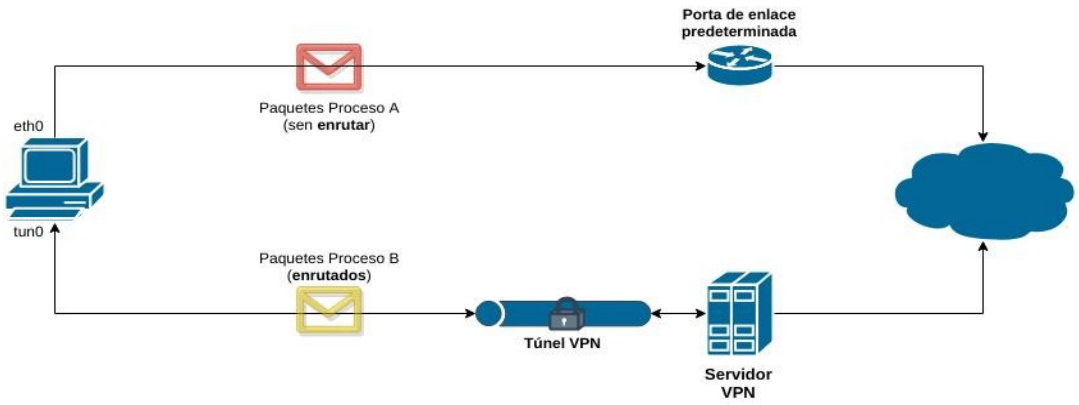

Ilustración 1: Esquema do sistema

É posible dificultar ser detectado se se manteñen ó mesmo tempo a conexión VPN e o tráfico regular, sen tunelar.

Para facilitar esto, desenvolveuse unha aplicación de escritorio que permite, de maneira sinxela, que os usuarios lancen as aplicacións que queiran enrutar a través dun VPN (ou, xeralizando, dunha interface calqueira).

Os programas lanzados a través da aplicación terán todo o seu tráfico enrutado por unha interface que se escolla (na figura de exemplo, *tun0*), mentres que o resto do tráfico do equipo emprega a tabla de rutas por defecto (na figura, tráfico de *eth0).*

Permitindo así ter tráfico enrutado e non enrutado simultáneamente, dificultando a detección do uso da VPN.

3. Resumo da solución proposta

Para cumprir os obxectivos descritos na sección anterior, a solución proposta consiste nunha aplicación con interface gráfica que busca ser o máis sinxela de empregar posible, para que calquer usuario poida beneficiarse dela, independentemente dos seus coñecementos de informática.

A aplicación consiste en:

- Menú lanzador de aplicacións
  - Menú que presenta tódalas aplicacións instaladas no sistema e ofrece a posibilidade de buscar por nombre
  - Explorador de ficheiros para buscar e lanzar ficheiros executables que non se atopen no menú
- Barra de administración
  - Configuración da aplicación
  - Botóns para iniciar e deter o enrutamento.

Para o desenvolvemento seguiuse unha variación sobre a metodoloxía do desenvolvemento iterativo incremental.

Debido a que ao comezo do desenvolvemento descoñecíase exactamente como íase a desenvolver a solución, considerouse que o máis axeitado sería dividir a solución en partes ata certo punto independientes, e investigar, deseñar, programar e probar cada parte da aplicación por separado.

Así, por exemplo, puideronse desenvolver os scripts de inicio e detención do enrutamento sen ter que preocuparse polo desenvolvemento da interface gráfica.

4. Planificación e seguimento

Ao comezo do proxecto, elaborouse una planificación a seguir durante tódolo desenvolvemento, en forma de diagrma de Gantt, no que se especifican as tareas a desenvolver semana a semana.

De igual maneira, o seguimento tamén se fixo semana a semana, anotando todo o progreso feito en cada a semana para posteriormente, ao final do proxecto, recopilalo todo nun diagrama de Gantt.

Nas seguintes páxinas amósanse ámbolos diagramas de Gantt.

Entre os dous diagramas pódense comprobar varias discrepancias:

No diagrama de seguimento existen tareas durante a etapa de Prototipo que non existen na planificación.

Isto é debido a que durante o desenvolvemento do prototipo tratouse de desenvolver unha solución que discriminase e enrutase o tráfico por porto de orixe. A aplicación buscaba mediante netstat os portos abertos polas aplicacións e aplicaba e eliminaba regras a medida que os procesos abrian e pechaban portos.

Esto resultou ser unha solución inefectiva, dado que as aplicacións abren e pechan portos aleatorios constantemente, entre que o programa buscaba os portos e aplicaba as regras, a aplicación xa cerraba os portos encontrados, polo que non funcionaba, obrigando a comezar de cero a investigación dende outro punto de vista.

Nesta segunda investigación, atopouse e desenvolveuse unha solución que discrimina por grupo principal. Na sección 8 pódese atopar unha descripción máis en detalle da solución.

Esta segunda investigación levou máis tempo de previsto por problemas persoais de saúde. Para axilizar o desenvolvemento comezou o desenvolvemento da interface de usuario (Reflexado nas tareas da etapa de Desenvolvemento do diagrama de seguimento) antes de finalizar o prototipo funcional da solución.

Unha vez finalizado tanto o prototipo coma a interface de usuario, integráronse ambas partes e finalmente podeuse desenvolver a aplicación final, que puido ser completada antes do previsto debido á sinxeleza e a abundancia de documentación de Python, Linux, e tódalas librerías empregadas no proxecto.

4.1 Planificación

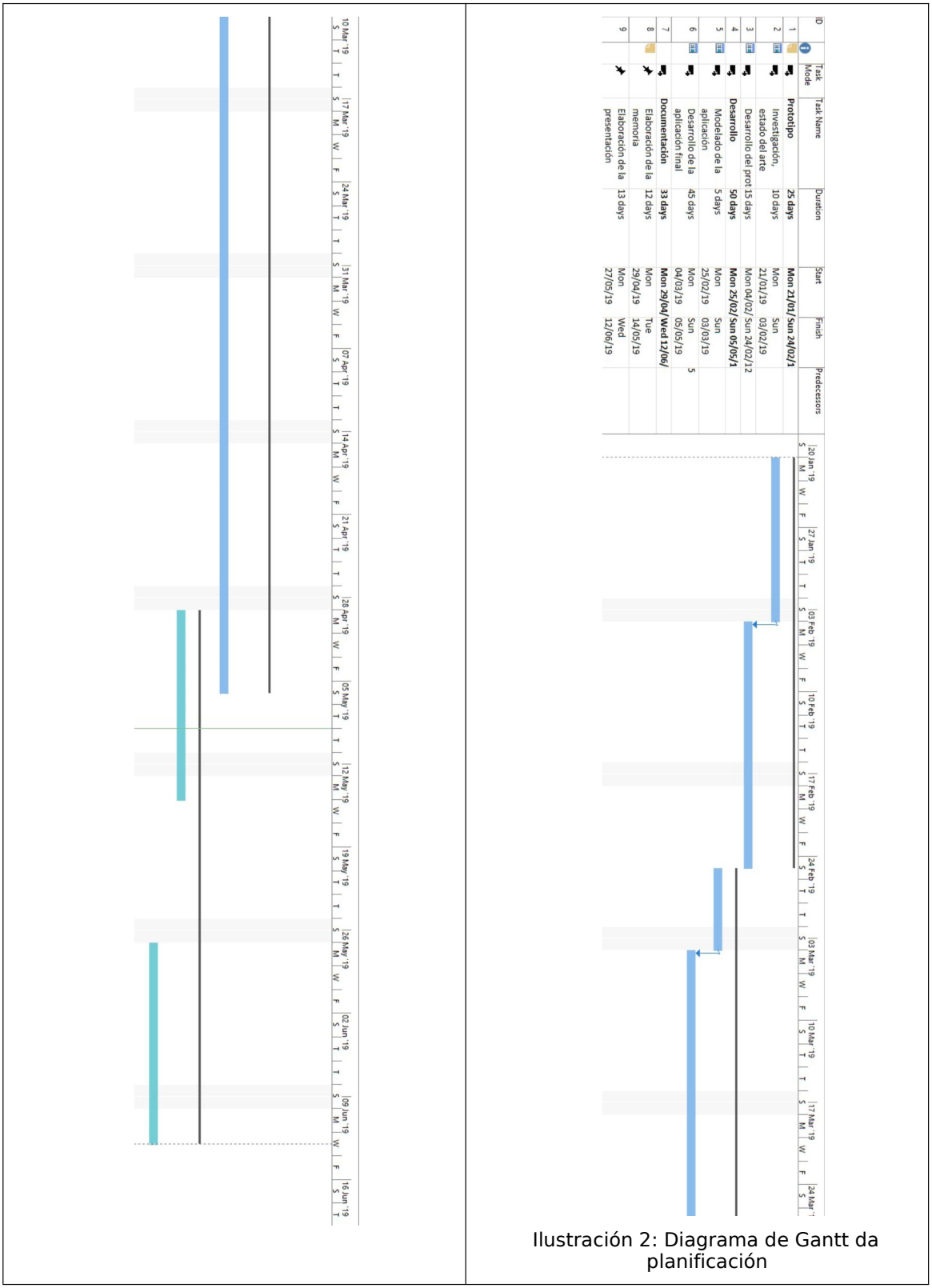

Ilustración 2: Diagrama de Gantt da planificación

4.2 **Seguimento**

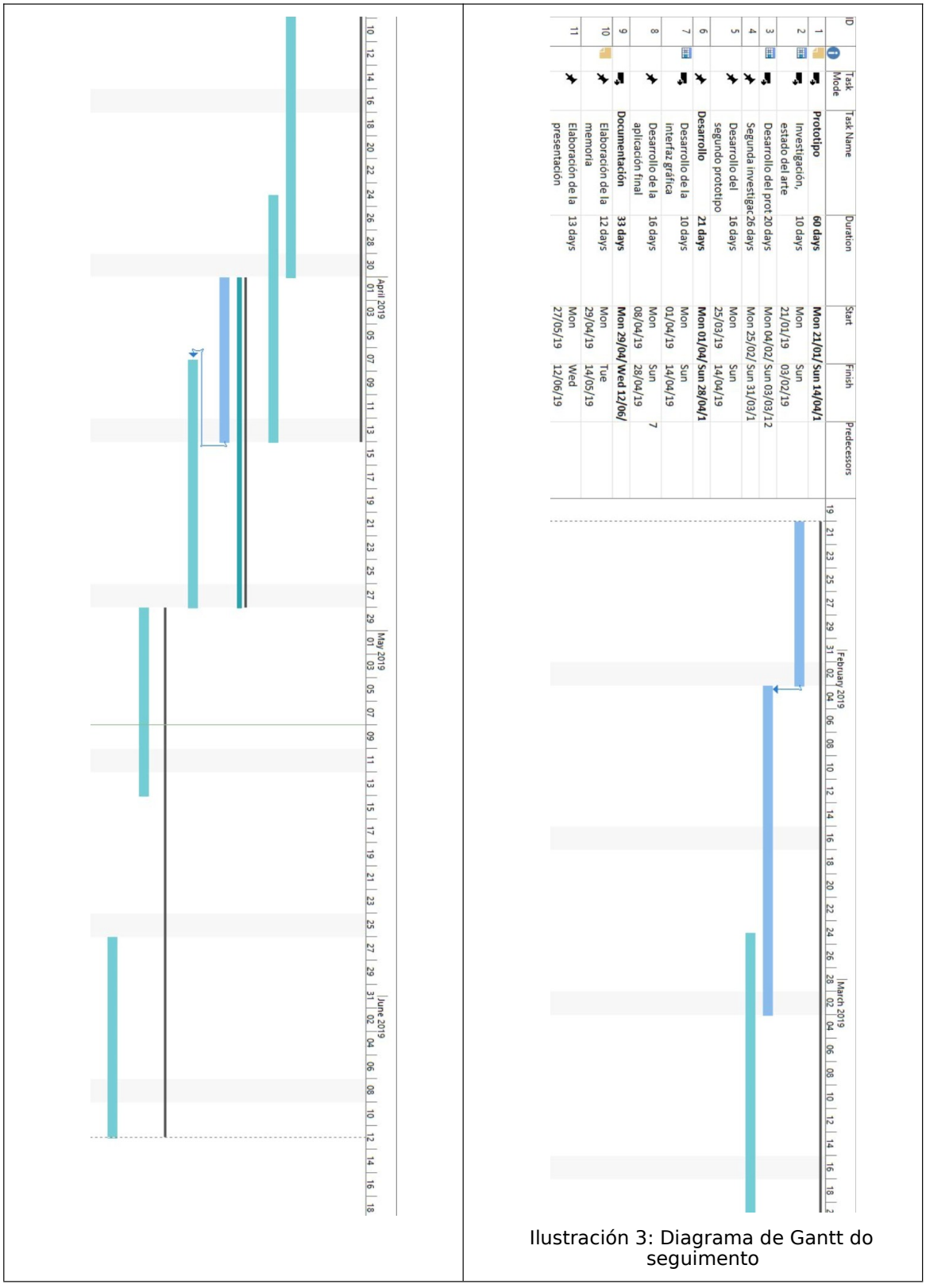

Ilustración 3: Diagrama de Gantt do seguimento

5. Arquitectura

Pódese dividir a aplicación en dúas capas:

Interface gráfica: Sigue unha arquitectura dirixida por eventos.

Cada ventana da aplicación é representada por unha clase que posée métodos para manexar eventos. No momento no que o usuario acciona algún dos elementos da interface, prodúcese un evento, e accionase método correspondente.

Os eventos poden modificar o estado da aplicación, alterar elementos da vista, lanzar novas vistas, ou accionar a segunda capa da aplicación

Scripts: Execútanse de maneira puntual ao ser accionados pola interface. Encárganse de interactuar co Sistema Operativo para configurar iptables e iproute.

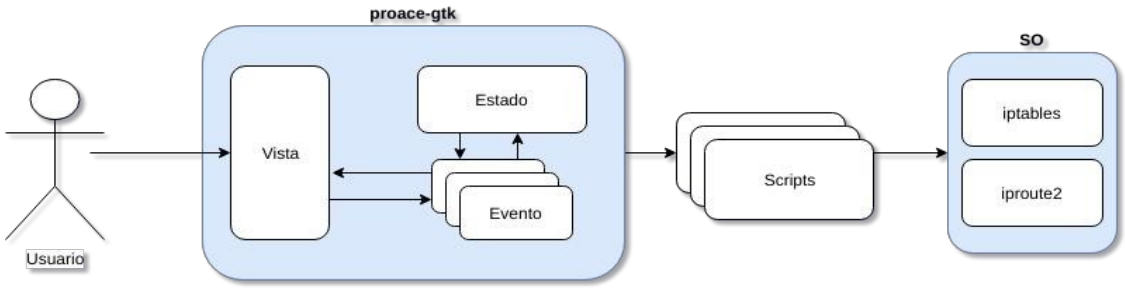

Ilustración 4: Arquitectura da solución

Na sección 8 explícase máis en detalle as funcións de cada unha das capas así como o seu funcionamento.

6. Tecnoloxías e integración de produtos de terceiros

Linux

A solución está fortemente baseada no funcionamento de **Linux**, en especial en **iptables e iproute2**. Non está destinada a ningúna distribución en específico e esperase que sexa compatible con toda distribución actual.

Como desvantaxe, o feito de que a aplicación esté tan fortemente baseada en Linux dificulta a súa portabilidade a outros Sistemas Operativos.

iptables e iproute2

Para o enrutamento e a manipulación de paquetes, o kernel de Linux ofrece as ferramentas de espacio de usuario iptables e iproute2.

A solución desenvolta fai uso de **iptables** para poder discriminar o tráfico dos procesos que se queren enrutar do resto do tráfico do sistema e **iproute2** para enrutar o tráfico discriminado cara a interface obxectivo mediante regras e táboas de rutas.

Explícase máis en detalle o uso destas ferramentas na sección 8.

VPN

Se ben é certo que os obxectivos mencionan explícitamente o uso de redes VPN, dado que a solución únicamente enruta cara a unha interface, sen ter en conta se esta interface é parte dun túnel VPN ou non, espérase tamén que calqueira tipo de VPN que empregue interfaces sexa compatible.

YAML

A aplicación garda a súa configuración nun ficheiro en formato YAML, unha linguaxe de serialización de datos legible por humanos.

Escolleuse este formato pola súa sinxeleza e pola súa facilidade de uso en Python empregando certas librerías.

GTK

GTK é unha colección de librerías multiplataforma que permite desenvolver interfaces gráficas.

Escolleuse principalmente por ser a librería empregada no desenvolvemento de GNOME (Entorno de escritorio de Ubuntu, a distribución de Linux máis popular), e pola abundancia de guías, tutoriales, e documentación, que facilitaron o desenvolvemento da interface gráfica.

Glade

Glade é unha ferramenta que permite definir visualmente plantillas para GTK.

Emprégase para definir declarativamente as vistas da interface gráfica.

Python3

Emprégase **Python3** coma linguaxe de programación, facendo uso de diversas librerías.

- **pyroute2:** Emprégase para interactuar con iproute2.
- **ruamel.yaml:** Permite traballar co ficheiro de configuración (en formato YAML)
- **python-gobject:** Aporta clases e métodos para traballar con GTK3. Emprégase para a interface gráfica

Bash

A interacción con iptables e iproute2 realízase a través de scripts en Bash. Emprégase pola súa sinxeleza para executar comandos por lotes.

Se ben é certo que os scripts de Bash poden dar problemas en casos excepcionais (coma, por exemplo, non especificar argumentos antes de lanzar o script), os scripts desenvoltos fan uso do Unofficial Bash Strict mode, para minimizar o risco de que se produzan situacións excepcionais problemáticas durante a execución dos scripts.

7. Especificación e análise de requisitos

7.1 Características dos usuarios

Deseñouse a solución tendo en mente ordenadores persoais, onde o usuario ou usuarios teñen permisos de administrador sobre o sistema. Porén, contémplase únicamente un tipo de usuario.

7.2 Funcións

A aplicación permite realizar as seguintes funcións:

- Iniciar enrutamento
- Deter enrutamento
- Modificar configuración
- Lanzar aplicación

7.3 Casos de uso

Seguindo as funcións especificadas na sección anterior, defínense os casos de uso mostrados na seguinte ilustración:

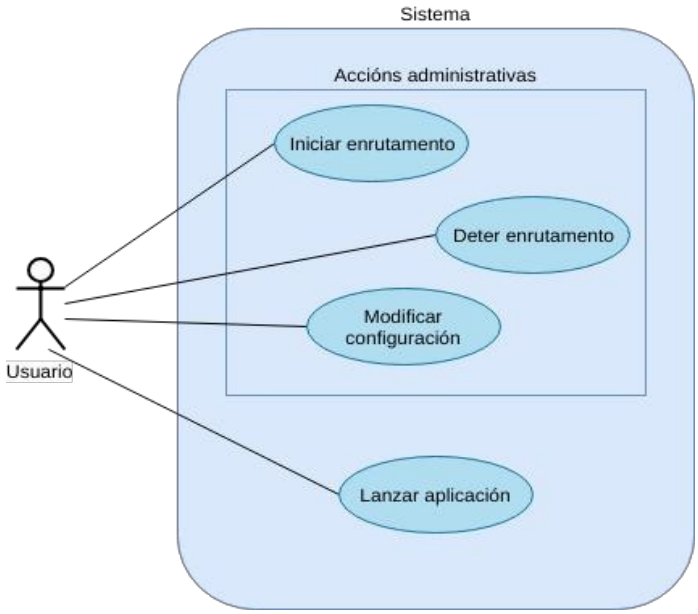

Ilustración 5: Diagrama de casos de uso

7.3.1 Accións administrativas

7.3.1.1 Iniciar servicio

Nombre	Iniciar enrutamento
Descrición	Configurar e engadir ao sistema regras e rutas de iproute2 e iptables para que as aplicacións lanzadas sexan enrutadas a través da interface obxectivo
Actor	Usuario
Precondición	A interface obxectivo debe estar activa O usuario debe ter permisos de administrador (root)
Poscondición	Configúrase e engádense as regras e rutas necesarias ao sistema

7.3.1.2 Deter servicio

Nombre	Deter enrutamento
Descrición	Eliminar as regras e rutas previamente engadidas e desfacer a configuración feita no sistema para o enrutamento
Actor	Usuario
Precondición	O enrutamento debeu haberse iniciado previamente O usuario debe ter permisos de administrador (root)
Poscondición	Desfanse os cambios feitos ao iniciar o enrutamento

7.3.1.3 Modificar configuración

Nombre	Modificar configuración
Descrición	Gardar os datos do menú de configuración ao ficheiro de configuración
Actor	Usuario
Precondición	O usuario debe ter permisos de administrador (root)
Poscondición	Os datos son almacenados no ficheiro

7.3.2 Accións regulares

7.3.2.1 Lanzar aplicación

Nombre	Lanzar aplicación
Descrición	Lanzar unha aplicación para que o seu tráfico sexa enrutado hacia a interface obxectivo
Actor	Usuario
Precondición	O enrutamento debeu haberse iniciado previamente
Poscondición	A aplicación escollida é lanzada e o seu tráfico e enrutado cara a interface obxectivo

8. Deseño do software

Ao longo desta sección describirase o deseño da aplicación, abordando por separado o deseño dos scripts que configuran o enrutamento e o deseño da interface gráfica.

8.1 Scripts

Podería considerarse que o núcleo da aplicación son os scripts que configuran o enrutamento; que permiten ao sistema distinguir que paquetes enrutar e cales non.

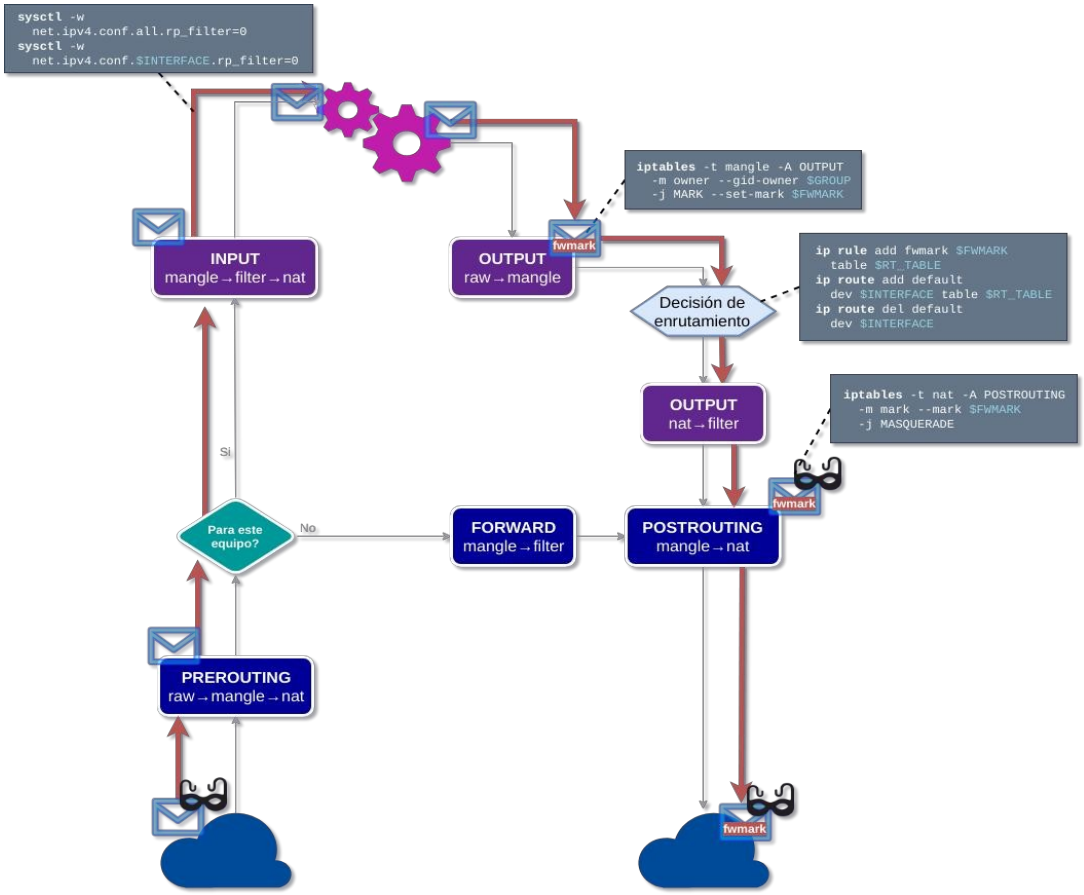

Ilustración 6: Seguimento dun paquete enrutado cara a interface obxectivo

Para enrutar os paquetes saíntes que proveñen dalgúns procesos en específico, é necesario atopar algunha forma de diferencialos dos paquetes provinientes dos demáis procesos.

Para eso **iptables** conta co módulo *owner*, que da opcións para diferenciar por usuario, ou grupo.

Para esta solución, escolleuse a opción de diferenciar por un grupo específico (en adiante, grupo obxectivo, e na ilustración *$GROUP*).

Deste xeito, mediante unha regra de iptables, os paquetes procedentes de procesos executados co grupo obxectivo como grupo principal son marcados cun número (en adiante, fwmark ou *$FWMARK*).

```
$ iptables -t mangle -A OUTPUT -m owner --gid-owner $GROUP -j MARK
--set-mark $FWMARK
```

Para que estes paquetes saian por unha interface en concreto (en adiante, interface obxectivo ou *$INTERFACE*) non chega con marcalos, tamén é necesario enrutalos:

Isto lógrase mediante unha regra de **ip rule**, para indicar que os paquetes marcados *deben ser enrutados mediante una táboa de rutas concreta* (en adiante, táboa de rutas obxectivo ou *$RT_TABLE*), que contén como porta de enlace por defecto a porta de enlace da rede da interface obxectivo.

Para evitar que o tráfico sexa tunleado pola VPN por defecto, elimínase da táboa de rutas por defecto a porta de enlace da interface obxectivo.

```
$ ip rule add fwmark $FWMARK table $RT_TABLE
$ ip route add default dev $INTERFACE table $RT_TABLE
$ ip route del default dev $INTERFACE
```

Ao forzar paquetes por esa porta de enlace, engádese un problema máis: Orixinariamente, esos paquetes ían dirixirse cara outra rede, polo que *teñen como IP de orixen unha IP que non corresponde cunha da interface obxectivo*.

Afortunadamente, é posible **enmascarar** os paquetes cunha IP que si corresponda á interface obxectivo mediante outra regra de iptables, que enmascare os paquetes marcados.

```
$ iptables -t nat -A POSTROUTING -m mark --mark $FWMARK -j MASQUERADE
```

Deste xeito, os paquetes de certos procesos en específico, que orixinariamente ían saír pola porta de enlace predeterminada do sistema, son redirixidos á rede da interface obxectivo, cunha IP correspondiente a dita interface.

Ao recibir unha resposta, iptables encárgase de desenmascarar os paquetes para que o sistema poida levar a resposta ao proceso que enviou a petición.

O sistema ve un problema con iso: dado que non espera recibir un paquete cunha IP e destino que non corresponde a unha asignada á interface obxectivo, descarta o paquete.

Isto coñécese como **Reverse Path Filtering**, e usualmente é algo desexable, mais, para este caso de uso, impídenos cumprir o noso obxectivo.

```
$ sysctl -w net.ipv4.conf.all.rp_filter=0
$ sysctl -w net.ipv4.conf.INTERFAZ.rp_filter=0
```

Unha vez desactivado, os procesos reciben as respostas aos seus paquetes.

Finalmente, tendo feita toda esta configuración, os paquetes procedentes de procesos executados co grupo obxectivo serán enrutados cara a interface de saída, mentres que os paquetes de tódolos demáis procesos non se ven afectados.

Así, os usuarios poden decidir que procesos son enrutados e cales non de forma sinxela: soamente teñen que executar os procesos a través do grupo obxectivo.

Unha forma de facer isto é mediante o comando sg:

```
$ sg grupo_obxectivo “/path/to/bin and arguments”
```

E outra forma sería a través da interface gráfica.

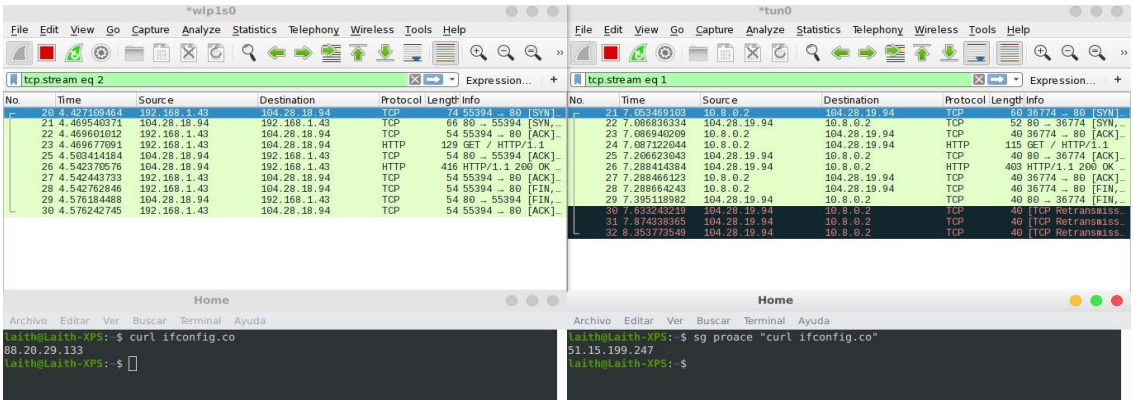

Ilustración 7: Captura de paquetes dun proceso sen enrutar (esquerda) e dun proceso enrutado (dereita)

Na ilustración 7 amósase o tráfico de dous procesos, un executado normalmente, e outro executado a través de sg para lanzalo co grupo obxectivo. Pódese comprobar como o tráfico é enrutado por interfaces distintas nos dous casos.

A aplicación posúe scripts que automatizan tódolos pasos descritos nesta sección, para que os usuarios poidan activar e desactivar o enrutamento de forma sinxela, sen ter que intervir máis alá da configuración inicial.

Estos scripts poden ser lanzados pola liña de comandos, ou preferiblemente, a través da interface gráfica.

Pódense atopar os scripts nos anexos 1 e 2.

8.2 Interface gráfica

A interface segue unha arquitectura dirixida por eventos.

Cada unha das clases representa unha vista, e cada clase posúe métodos para responder aos eventos xerados polas accións do usuario na interface.

As vistas, ao ser instanciadas, cargan a súa plantilla correspondente dun ficheiro e constrúen unha nova ventana cos contidos da plantilla.

Dado a que a aplicación presenta un número moi reducido de casos de uso, a interface de usuario non precisa ser moi complexa para cubrir tódolos casos de uso.

Esta arquitectura pode ser resumida no seguinte diagrama de clases:

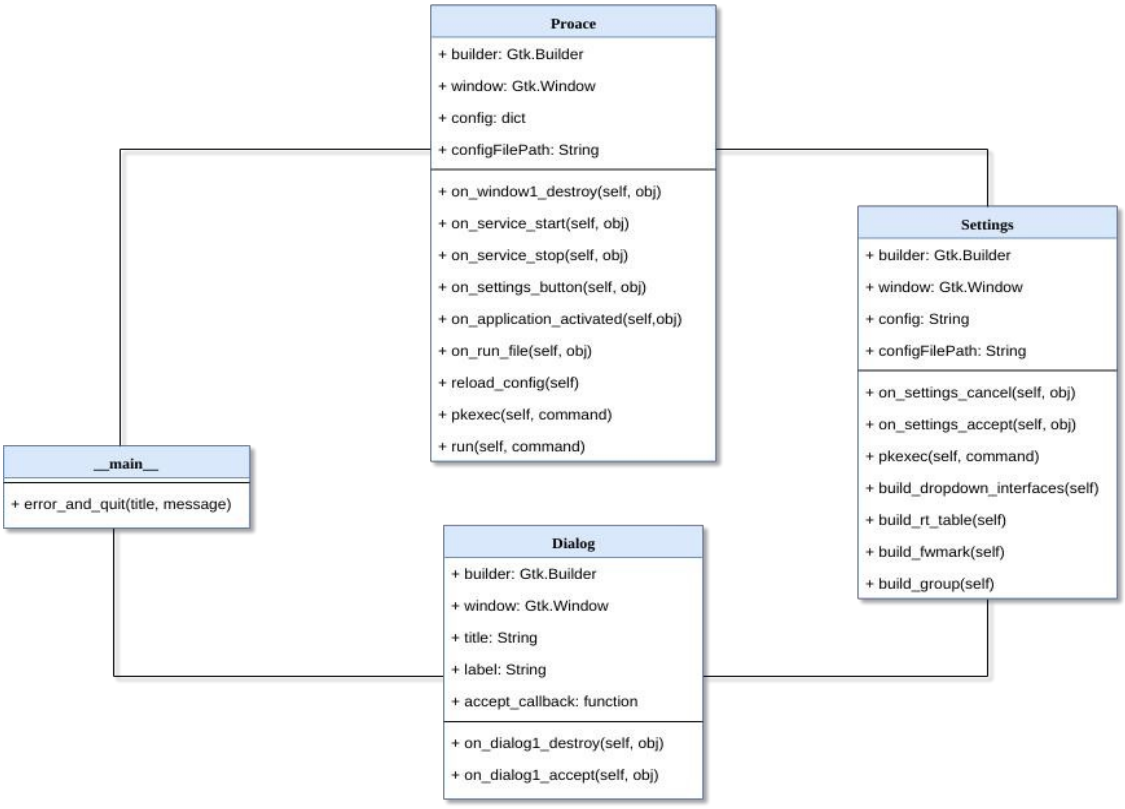

Ilustración 8: Diagrama de clases

Nas seguintes seccións explicaranse algúns dos métodos máis importantes da aplicación.

8.2.1 Proace

8.2.1.1 pkexec

Tamén presente na clase Settings, iste método, que toma un array de argumentos, executa coma root os argumentos do array a través de *pkexec* para pedir cunha interface gráfica os permisos de root.

Iste método emprégase en métodos manexadores de eventos coma on_service_start e on_service_stop para executar os scripts da aplicación, explicados na sección 8.2, con permisos de root.

8.2.1.2 run

Iste método toma un array de argumentos e os executa a través do grupo obxectivo (especificado no ficheiro de configuración).

É equivalente a executar na liña de comandos o seguinte:

```
$ sg grupo_obxectivo “array de argumentos”
```

Emprégase para lanzar aplicacións a través de interface grafica como o grupo obxectivo, para que o seu tráfico sexa enrutado.

8.2.2 Settings

8.2.2.1 on_settings_accept

Actúa como manexador do evento on_settings_accept, accionado na vista de *Settings* cando o usuario oprime o botón de aceptar para salvar as preferencias.

Recopila nun array asociativo, ou dicionario (*dict()*) os datos presentes na vista de Settings e posteriormente tenta escribilos en formato *YAML* no ficheiro de configuración (atributo *configFilePath*).

No caso de non ter permisos de escritura, chámase ao método *pkexec* para tentar de escribir as preferencias como root.

9. Xestión de datos e información

Os únicos datos que son manexados pola aplicación son os datos do ficheiro de configuración, en formato *YAML*.

O usuario da aplicación pode indicar a ruta do ficheiro de configuración como argumento ao lanzar a aplicación. No caso de omitilo, o programa buscará un ficheiro "proace.yaml" no directorio de traballo e, se non o encontra, o buscará en /etc/proace.yaml. No caso de non atopalo en ningunha ubicación, o programa pecharase amosando un mensaxe de erro.

O ficheiro, pode ser modificado a man, dado que é texto plano, ou pode ser modificado na propia aplicación na sección de Preferencias. Ao gardar os cambios no menú, a aplicación gardará os datos na ruta na que se atopou o ficheiro ao lanzar o programa.

Contidos de exemplo do ficheiro *proace.yaml*:

```
{
   fwmark: 33,
   group: proace,
   interface: tun0,
   rt_table: 33
}
```

10. Probas levadas a cabo

As probas realizaronse nun sistema **Ubuntu 18.04 LTS** no que se instalou a aplicación seguindo as indicacións da sección 11.

O sistema conectouse a un perfil de *OpenVPN* escollido ao azar de entre os dispoñibles libremente en vpngate.net para demostrar que a aplicación pode funcionar con calqueira VPN.

Para probar o correcto funcionamiento do enrutamento por aplicación utilizaronse dous navegadores: **Firefox** e **Chromium.**

O primeiro lanzouse de forma regular para que o seu tráfico no sexa enrutado, e o segundo lanzouse a través da aplicación para que o seu trafico sexa enrutado.

Os dous navegadores conectaronse ao sitio web ifconfig.co, que indica a IP pública do cliente que se conecta.

En adelante, enténdese por "IP habitual" a IP que se amosaría no sitio web sen conectarse á VPN (No esquema, *83.165.248.157*) e por "IP da VPN" a IP que se amosaría no sitio web ao conectarse á VPN (No esquema, *51.15.199.247*).

Na seguinte ilustración amósase un esquema da rede para este exemplo:

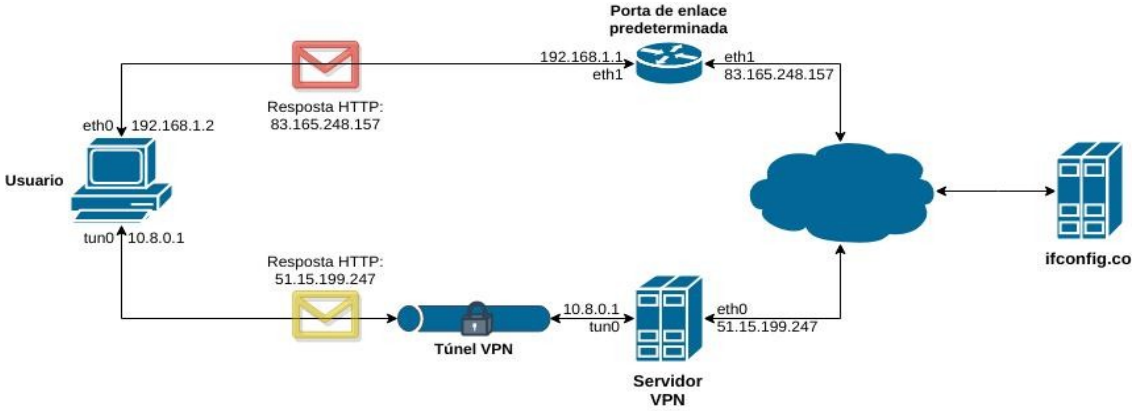

Ilustración 9: Esquema da rede para este exemplo, amosando interfaces e direccións IP

Nas seguintes taboas se explican todas as probas levadas a cabo, indicando o resultado esperado e o resultado obtido. Considérase que una proba é exitosa se ámbolos resultados coinciden.

Proba	Coa VPN funcionando, iniciar o enrutamento na interface da VPN.
Resultado esperado	Firefox debe mostrar a IP pública habitual. Chromium debe mostrar a IP pública da VPN.
Resultado obtido	Firefox mostra a IP pública habitual. Chromium mostra a IP pública da VPN.
Éxito	OK

Proba	Coa VPN funcionando e o enrutamento iniciado na interface da VPN, deter o enrutamento.
Resultado esperado	Tanto Firefox como Chromium deben mostrar a IP da VPN
Resultado obtido	Ámbolos navegadores mostran a IP da VPN
Éxito	OK

Proba	Iniciar o enrutamento nunha interface inexistente ou inicialo coa VPN desconectado
Resultado esperado	Firefox debe mostrar a IP habitual. Chromium non debe ser capaz de conectarse a internet
Resultado obtido	Ámbolos navegadores mostran a IP habitual.
Éxito	Fallo

Proba	Coa VPN conectado, deter o enrutamento sen habelo iniciado primeiro
Resultado esperado	Firefox debe mostrar a IP da VPN. Chromium non debe ser capaz de conectarse a internet
Resultado obtido	Ámbolos navegadores mostran a IP da VPN.
Éxito	Fallo

Proba	Coa VPN funcionando e o enrutamento iniciado na interface da VPN, iniciar o enrutamento de novo.
Resultado esperado	Firefox debe mostrar a IP habitual. Chrome debe mostrar a IP da VPN. Es decir, debe funcionar como se sóamente su iniciase unha única vez
Resultado obtido	Firefox mostra a IP habitual. Chrome mostra a IP da VPN.
Éxito	OK

11. Manual de usuario

Nesta sección asúmese que o usuario posúe un sistema Ubuntu 18.04 LTS. É posible que algúns dos pasos poidan variar lixeiramente entre distribución e distribución.

11.1 Requirimentos

Para o correcto funcionamento da aplicación, é preciso ter instalados no sistema os seguintes paquetes:

- python3
- python3-gobject
- python3-pyroute2
- python3-ruamel.yaml

11.2 Instalación

1. Copiar os contenidos do directorio *proace-gtk* en calqueira ubicación do sistema.
2. Engadir ao sistema un novo grupo de usuarios (Precisa permisos de root)

```
# groupadd proace
```

3. Engadir ao novo grupo tódolos usuarios do sistema que queiran utilizar a aplicación (Precisa permisos de root)

```
# usermod -a -G proace USUARIO
```

Reemplazando “USUARIO” polo usuario que se queira engadir ao grupo.

11.3 Manual de uso

Antes de comezar con esta sección, asegúrese de que a interface obxectivo está activa ou conéctese á VPN en caso de que desexe usar unha, e execute *proace-gtk.*

```
$ ./proace-gtk
```

Opcionalmente, pode indicar nos argumentos unha ruta a un ficheiro de configuración.

Se non o indica, a aplicación buscará a configuración nos seguintes ficheiros, en orden:

- ./proace.yaml
- /etc/proace.yaml

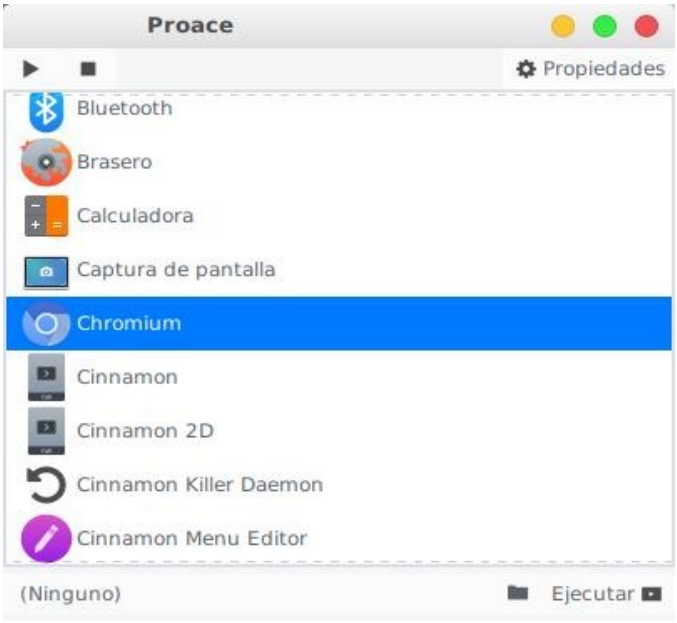

Ilustración 10: Vista principal da interface gráfica

A vista principal de Proace pode dividirse en dúas partes:

- Barra de administración

  É a barra superior da aplicación conten os botóns de inicio e detención do enrutamento (Botóns de *play* e *stop*) así como o acceso ao menú de configuración ou *Propiedades*.

- Lanzador de aplicacións

  Presenta unha lista con tódalas aplicacións instaladas no sistema así como a opción de buscar por nome se o usuario teclea.

  É posible lanzar executables que non se encontren na lista seleccionandoos nun explorador de ficheiros que se lanza co botón da barra inferior do lanzador e executando facendo click no botón de *Ejecutar*.

11.3.1 Configuración

E moi probable que precise configurar a aplicación antes de poder usala. Proace por defecto enruta cara a interface tun0.

Se desexa cambiar os parámetros, pode facer os cambios no ficheiro de configuración ou no menú de Propiedades.

Ilustración 11: Vista da configuración.

Neste menu, pódese escoller a interface obxectivo nun menú desplegable.

Tamén é posible cambiar os demáis parámetros, explicados máis en detalle na *sección 8*, aínda que normalmente non é necesario.

11.3.2 Inicio e detención do enrutamento

- Para iniciar o enrutamento: Oprima o botón de "*Play*" (►).
- Para deter o enrutamento: Oprima o botón de "*Stop*" (■).

Hai que ter en conta que o enrutamento aplícase sobre a configuración actual no momento de realizar as accións.

É dicir, se tras iniciar o enrutamento, o usuario cambia as preferencias, os cambios non se verán reflexados ata que se reinicie o enrutamento.

11.3.3 Lanzamento de aplicacións

Pódense lanzar aplicacións de dúas formas:

- Seleccionando mediante doble click unha aplicación no menú de aplicacións
- Seleccionando un ficheiro executable no explorador de ficheiros e premendo no botón "*Ejecutar*".

Todo o tráfico das aplicacións lanzadas será redirixido cara a interface obxectivo sempre e cando o enrutamento estéa iniciado.

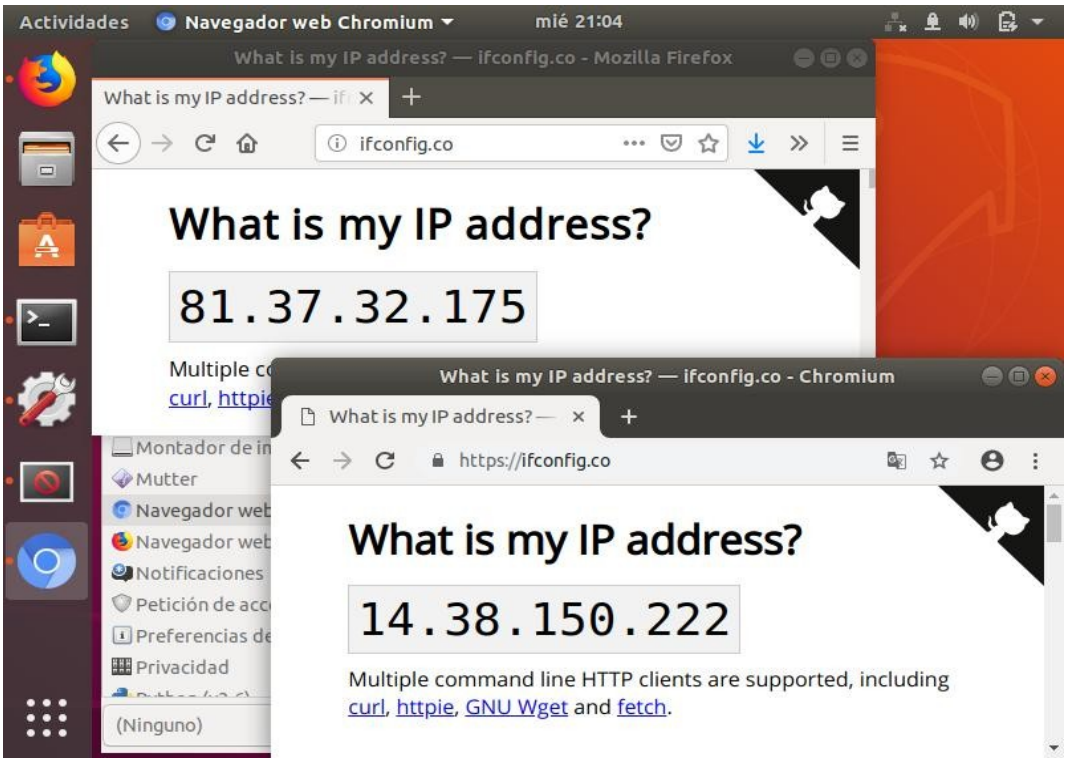

Ilustración 12: Firefox (esquerda), sen enrutar, amosando a IP pública regular e Chromium (dereita), enrutado, amosando a IP pública da VPN

11.4 Uso a través da liña de comandos

Aínda que non é o recomendado, é posible realizar tódalas accións da aplicación sen empregar a interface gráfica, a través da liña de comandos.

11.4.1 Inicio e detención do enrutamento

No directorio *proace_sudo* atopará os dous ficheiros responsables de iniciar e deter o enrutamento: *start.sh* e *stop.sh*. Os dous scripts toman os mesmos argumentos no mesmo orde:

```
$ start.sh INTERFACE RT_TABLE FWMARK GROUP
```

Os argumentos correspóndense con a *interface obxectivo, taboa de rutas obxectivo, fwmark* e *grupo obxectivo* explicados na sección 8.1.

11.4.2 Lanzamento de aplicacións

Mediante o comando *sg* é posible executar procesos con un grupo principal diferente ao grupo principal do usuario que executa o proceso.

Debido a que a solución permite ao sistema separar o tráfico segundo o grupo principal dos procesos (máis detalles na sección 8.1), o tráfico dos procesos lanzados con *sg* co grupo obxectivo será enrutado cara a interface obxectivo.

```
$ sg GROUP "/path/to/bin and arguments"
```

Sendo *GROUP* o grupo obxectivo e */path/to/bin and arguments* o comando a executar, cos seus argumentos en caso de precisalos.

12. Principais aportacións

Na sección 2 definironse uns obxectivos que surxiron a partir da necesidade de "esconder" o tráfico dun túnel VPN entre tráfico normal, e a solución cumpre os obxectivos:

- Manter simultáneamente tráfico dirixido a internet enrutado cara a un túnel VPN e tráfico sen enrutar, con configuración mínima.

Pero a maiores, tamén fai as seguintes aportacións:

- Facilidade para discriminar o tráfico de procesos determinados.
- Capacidade de enrutar todo o tráfico das aplicacións discriminadas cara a unha interface en específico.
- Sinxeleza e facilidade de uso da aplicación, permitindo que poida ser utilizada por usuarios non experimentados ou con poucos coñecementos de redes e do funcionamiento de Linux e iptables.

Todas estas aportacións poden dar lugar a vías de traballo futuro, como as que se amosarán na sección 14.

13. Conclusións

En retrospectiva, aínda que a aplicación final resultou ser relativamente sinxela debido ao seu número reducido de casos de uso, o seu desenvolvemento supuxo un importante desafío.

Ao empezar o proxecto, descoñecía totalmente o funcionamento de iptables e os meus coñecementos sobre cómo funcionaba o enrutamento en Linux eran limitados, polo que as primeiras etapas do desenvolvemento resultaron accidentadas. En múltiples ocasións vinme obrigado a replantexar cómo sería a solución ata que finalmente atopei unha solución funcional e puiden por fin proseguir co desenvolvemento da aplicación final.

Sobre o uso de Python, Glade, GTK, e outras librerías para a interface, a pesar de ser a primeira vez que empregaba estas tecnoloxías, non supuxeron un problema maior, gracias a tódala documentación dispoñible libremente en internet. Resultou ser sorprendentemente sinxela a elaboración da interface.

Finalmente, se ben e certo que a solución desenvolta cumpre cos obxectivos e é funcional, compre destacar que se trata dun prototipo que non foi probado en ningún entorno, real ou simulado, no que se poidese determinar con seguridade se é efectivo para “camuflar” o tráfico enrutado a través dun túnel VPN. Polo que se desaconsella o uso da aplicación en calquera situación no que o uso dun VPN poida supoñer un risco. O software deste traballo non inclúe ningunha garantía.

14. Vías de traballo futuro

Tendo en conta as aportacións mencionadas na sección 12, pódense idear moitas melloras e ampliacións posibles que poden dar lugar a novas vías de traballo futuro:

A solución desenvolta aporta unha forma de discriminar o tráfico de procesos determinados, polo que pode servir como base para un firewall que filtre por procesos, unha das aplicacións de Linux máis demandadas que todavía non existe.

Tamén podería extenderse a solución para permitir manexar múltiples perfís, permitindo enrutar alguns procesos cara a unha interface, e outros distintos cara outra interface, ou incluso bloquear a conexión a internet aos procesos que se queiran.

Outra posibilidade sería a elaboración de software que cumpra os mesmos obxectivos pero para distintas plataformas, coma Mac OS, Windows, Android, etc.

Windows ofrece un firewall capaz de discriminar por executable, polo que a discriminación por procesos pode chegar a ser máis sinxela nesa plataforma.

Android, que fai uso do kernel de Linux, executa cada aplicación con un usuario distinto, polo que podería implementarse una solución similar que discrimine por usuario en lugar de por grupo.

15. Referencias

[1] «iptables(8) - Linux man page». [Online]. Dispoñible en: https://linux.die.net/man/8/iptables. [Accedido: 06-may-2019]

[2] «linux - Block network access of a process?», Unix & Linux Stack Exchange. [Online]. Dispoñible en: https://unix.stackexchange.com/questions/68956/block-network-access-of-a-process/69017. [Accedido: 06-may-2019]

[3] «linux - Output traffic on different interfaces based on destination port», Unix & Linux Stack Exchange. [Online]. Dispoñible en: https://unix.stackexchange.com/questions/21093/output-traffic-on-different-interfaces-based-on-destination-port/21118. [Accedido: 06-may-2019]

[4] J. Thornton, «Glade3 Python GTK Tutorial», gnipsel, 2012. [Online]. Dispoñible en: https://www.gnipsel.com/glade/glade01a.html

[5] «4.8. Routing Tables». [Online]. Dispoñible en: http://linux-ip.net/html/routing-tables.html. [Accedido: 06-may-2019]

[6] «D.3. ip rule». [Online]. Dispoñible en: http://linux-ip.net/html/tools-ip-rule.html. [Accedido: 06-may-2019]

[7] «D.2. ip route». [Online]. Dispoñible en: http://linux-ip.net/html/tools-ip-route.html. [Accedido: 06-may-2019]

[8] «ubuntu - How does linux decide the interface to route an application's traffic from?», Unix & Linux Stack Exchange. [Online]. Dispoñible en: https://unix.stackexchange.com/questions/420400/how-does-linux-decide-the-interface-to-route-an-applications-traffic-from/420436. [Accedido: 06-may-2019]

[9] «The Python GTK+ 3 Tutorial — Python GTK+ 3 Tutorial 3.4 documentation». [Online]. Dispoñible en: https://python-gtk-3-tutorial.readthedocs.io/en/latest/. [Accedido: 06-may-2019]

[10] «ruamel.yaml — Python YAML package documentation». [Online]. Dispoñible en: https://yaml.readthedocs.io/en/latest/. [Accedido: 06-may-2019]

[11] «Pyroute2 netlink library — pyroute2 0.5.5 documentation». [Online]. Dispoñible en: https://docs.pyroute2.org/. [Accedido: 06-may-2019]

[12] «GTK+ 3 Reference Manual: GTK+ 3 Reference Manual». [Online]. Dispoñible en: https://developer.gnome.org/gtk3/stable/. [Accedido: 06-may-2019]

[13] «Use the Unofficial Bash Strict Mode (Unless You Looove Debugging)». [Online]. Dispoñible en: http://redsymbol.net/articles/unofficial-bash-strict-mode/. [Accedido: 09-may-2019]

16. Anexos

16.1 Anexo 1: Script de inicio do enrutamento (start.sh)

```
#!/bin/bash
# Unofficial bash strict mode
set -euo pipefail
IFS=$'\n\t'

INTERFACE=$1
RT_TABLE=$2
FWMARK=$3
GROUP=$4

# Desactivar Reverse Path Filtering
sysctl -w net.ipv4.conf.all.rp_filter=0
sysctl -w net.ipv4.conf.$INTERFACE.rp_filter=0

# Marcar paquetes provinientes de procesos ejecutados con el grupo objetivo
iptables -t mangle -A OUTPUT -m owner --gid-owner $GROUP -j MARK --set-mark $FWMARK

# Regla para dirigir los paquetes marcados a la tabla de rutas especificada
ip rule add fwmark $FWMARK table $RT_TABLE

# En la tabla de rutas: Los paquetes se enrutarán por la puerta de enlace de la interfaz objetivo
ip route add default dev $INTERFACE table $RT_TABLE

# Eliminar puerta de enlace de la interfaz objetivo de la tabla de rutas principal
ip route del default dev $INTERFACE

# Enmascarar la IP de los paquetes marcados con una IP propia de la interfaz objetivo
iptables -t nat -A POSTROUTING -m mark --mark $FWMARK -j MASQUERADE
```

16.2 Anexo 2: Script de detención do enrutamento (stop.sh)

```
#!/bin/bash
# Unofficial bash strict mode
set -euo pipefail
IFS=$'\n\t'

INTERFACE=$1
RT_TABLE=$2
FWMARK=$3
GROUP=$4

# Limpiar las reglas de iptables establecidas
iptables -t mangle -D OUTPUT -m owner --gid-owner $GROUP -j MARK
--set-mark $FWMARK
iptables -t nat -D POSTROUTING -m mark --mark $FWMARK -j MASQUERADE

# Limpiar la tabla de rutas
ip route flush table $RT_TABLE

# Limpiar reglas de ip rule
ip rule del fwmark $FWMARK table $RT_TABLE

# Restablecer puerta de enlace por la interfaz objetivo en la tabla
# de rutas principal
ip route add default dev $INTERFACE

# No se reactiva rp_filter por si otra aplicación lo necesita
```