

---

# Informe de Vulnerabilidades de Quipux

Ola Bini, Analista Principal de Seguridad

2025-11-01



# Índice

<b>1</b>	<b>Introducción</b>	<b>1</b>
<b>2</b>	<b>¿Qué es Quipux?</b>	<b>2</b>
<b>3</b>	<b>Vulnerabilidades</b>	<b>3</b>
3.1	SQL Injection . . . . .	3
3.1.1	Inyección SQL en la funcionalidad de búsqueda (sql i .1) . . . . .	4
3.1.2	Inyección SQL en la funcionalidad de búsqueda 2 (sql i .2) . . . . .	4
3.1.3	Inyección SQL en adjuntos (sql i .3) . . . . .	4
3.1.4	Inyección SQL en formularios de administración (sql i .4) . . . . .	4
3.1.5	Inyección SQL en formularios de administración 2 (sql i .5) . . . . .	4
3.1.6	Inyección SQL en formularios de administración 3 (sql i .6) . . . . .	4
3.1.7	Inyección SQL en documentos asociados (sql i .7) . . . . .	5
3.1.8	Inyección SQL en documentos asociados 2 (sql i .8) . . . . .	5
3.1.9	Inyección SQL en documentos asociados 3 (sql i .9) . . . . .	5
3.1.10	Inyección SQL en documentos asociados 4 (sql i .10) . . . . .	5
3.1.11	Inyección SQL en la funcionalidad de ubicación (sql i .11) . . . . .	5
3.1.12	Inyección SQL en la funcionalidad de ubicación 2 (sql i .12) . . . . .	5
3.1.13	Inyección SQL en documentos asociados 4 (sql i .10) . . . . .	6
3.1.14	Inyección SQL en la funcionalidad de ubicación (sql i .11) . . . . .	6
3.1.15	Inyección SQL en la funcionalidad de ubicación 2 (sql i .12) . . . . .	6
3.1.16	Inyección SQL en la funcionalidad de ubicación 3 (sql i .13) . . . . .	6
3.1.17	Inyección SQL en la funcionalidad de ubicación 4 (sql i .14) . . . . .	6
3.1.18	Inyección SQL en la funcionalidad de ubicación 5 (sql i .15) . . . . .	6
3.1.19	Inyección SQL en informes (sql i .16) . . . . .	6
3.1.20	Inyección SQL en transacciones (sql i .17) . . . . .	7
3.1.21	Inyección SQL en transacciones 2 (sql i .18) . . . . .	7
3.1.22	Inyección SQL en las transacciones 3 (sql i .19) . . . . .	7
3.1.23	Inyección SQL en las transacciones 4 (sql i .20) . . . . .	7
3.1.24	Inyección SQL en las transacciones 5 (sql i .21) . . . . .	7
3.1.25	Inyección SQL en las transacciones 6 (sql i .22) . . . . .	7

3.1.26	Inyección SQL en las transacciones 7 (sql i .23) . . . . .	7
3.1.27	Inyección SQL en la carga de archivos (sql i .24) . . . . .	8
3.2	Cross-site scripting - XSS (xss .1) . . . . .	8
3.3	Divulgación de datos (info .1) . . . . .	8
<b>4</b>	<b>Recomendaciones</b>	<b>9</b>
<b>5</b>	<b>Cronología</b>	<b>10</b>
5.1	Semana del 28 de abril de 2025 . . . . .	10
5.2	3 de mayo de 2025 . . . . .	10
5.3	9 de mayo de 2025 . . . . .	10
5.4	30 de mayo de 2025 . . . . .	10
5.5	12 de junio de 2025 . . . . .	10
5.6	16 de junio de 2025 . . . . .	11
5.7	17 de junio de 2025 . . . . .	11
5.8	18 de junio de 2025 . . . . .	11
5.9	20 de junio de 2025 . . . . .	11
5.10	12 de agosto de 2025 . . . . .	11
5.11	4 de septiembre de 2025 . . . . .	11
5.12	10 de septiembre de 2025 . . . . .	11
5.13	16 de septiembre de 2025 . . . . .	12
5.14	17 de septiembre de 2025 . . . . .	12
5.15	18 de septiembre de 2025 . . . . .	12
5.16	22 de octubre de 2025 . . . . .	12
5.17	1 de noviembre de 2025 . . . . .	12
<b>6</b>	<b>Referencias</b>	<b>13</b>

# 1 Introducción

Este informe técnico describe varias vulnerabilidades encontradas en Quipux por el equipo de Seguridad Digital EC. Estas vulnerabilidades se identificaron mediante el análisis del código fuente públicamente disponible. Si bien existen ciertas contramedidas implementadas, no son suficientes para proteger contra diversas condiciones problemáticas. El equipo de Seguridad Digital EC practica una divulgación responsable y ha dado al proveedor tiempo suficiente para solucionar los problemas identificados antes de su publicación.

Este informe describe 24 casos de inyección SQL. Estas vulnerabilidades sólo pueden explotarse tras la autenticación, pero debido a la naturaleza abierta de las inyecciones, cualquiera de ellas compromete la base de datos completa. Este informe también describe un caso de cross-site scripting reflejado (XSS) que puede utilizarse para comprometer cuentas de usuario. Finalmente, también se encontró un caso de divulgación de datos.

Las vulnerabilidades identificadas han sido asignadas como CVE-2025-55341 para la vulnerabilidad XSS, CVE-2025-55342 para la divulgación de información y CVE-2025-55343 para las 24 inyecciones SQL. Al momento de la publicación, estas vulnerabilidades aún se encuentran reservadas.

Seguridad Digital EC sigue los estándares de divulgación responsable de Google Project Zero. Esto implica otorgar al proveedor (el Ministerio de Telecomunicaciones y de la Sociedad de la Información) 90 días a partir de la notificación inicial para corregir cualquier problema identificado. Al momento de la publicación de este informe, varias de las vulnerabilidades se han corregido, mientras que otras siguen activas.

El código fuente original analizado se publicó en <https://minka.gob.ec/quipux-comunitario/quipux-comunitario>, pero durante el proceso de corrección, el proveedor eliminó este repositorio. El nuevo código fuente se puede encontrar en <https://minka.gob.ec/mintel/ge/quipux/quipuxcomunitario>. Sin embargo, como referencia, el código fuente original analizado se puede encontrar en <https://github.com/seguridaddigitalec/quipux-old>.

El equipo de Seguridad Digital EC desea agradecer oficialmente a Fluid Attacks por su colaboración en el reporte de las vulnerabilidades CVE en cuestión. Asimismo, expresamos nuestro agradecimiento al equipo del Ministerio de Telecomunicaciones por su apertura y disposición para recibir nuestro informe y corregir los problemas identificados.

## 2 ¿Qué es Quipux?

Quipux es un sistema de gestión documental para el sector público desarrollado en Ecuador, basado en un código fuente existente de Colombia. El sistema es Software Libre y está disponible públicamente para su revisión. Varias instituciones públicas en Ecuador utilizan versiones de Quipux, incluyendo el gobierno central y muchas universidades. El sistema es mantenido por el Ministerio de Telecomunicaciones y de la Sociedad de la Información. No está completamente claro qué versiones de Quipux se encuentran implementadas en los diferentes entornos. El código fuente disponible públicamente no se ha actualizado durante algún tiempo y parece que las versiones instaladas tienen funcionalidades que no son visibles en el código fuente. Por esta razón, es posible que el desarrollo haya continuado de forma privada y que el código fuente más reciente no esté disponible públicamente (aunque esto contravenga la licencia del software y las leyes y regulaciones de Ecuador).

Estimaciones conservadoras nos llevan a creer que existen al menos 20 instalaciones, pero es más probable que muchas más instalaciones existan en el país. Es bien sabido que el sistema es utilizado a diario por casi todos los funcionarios del sector público, lo que lo convierte en un sistema con cientos de miles de usuarios diarios. Si bien no se sabe con certeza qué código se ejecuta en algunas de estas instituciones, parece probable que la mayoría de las vulnerabilidades encontradas se conserven en estas instalaciones.

Dado que los sistemas oficiales de registro, gestión y mantenimiento de datos y firma digital se ejecutan dentro de Quipux, es innegable la importancia de este sistema para la administración del país. Cualquier tipo de vulnerabilidad debe considerarse extremadamente grave.

## 3 Vulnerabilidades

Todas las vulnerabilidades documentadas en este informe se identificaron en el commit e1774acd75e4c538413a137304ccee8fba7b138 ((que corresponde al último del repositorio antiguo). Sin embargo, se ha determinado que todas las vulnerabilidades estaban presentes también en versiones anteriores, que se remontan a varios años atrás. Todas las vulnerabilidades han sido corregidas en la versión más reciente de la base de código.

### 3.1 SQL Injection

Se encontraron 24 instancias de inyección SQL en el código fuente. Estas se encuentran registradas como CVE-2025-55343. Estas corresponden a CWE-89 y CWE-74. La causa raíz de las inyecciones SQL ocurre cuando una query a la base de datos se realiza de tal manera que los datos controlados por un usuario final se componen con la query de tal manera que su semántica cambia. Esto puede tener diversos impactos, dependiendo de las demás protecciones existentes. En el caso más extremo, se puede extraer todo el contenido de la base de datos y modificarlo. En ciertos casos, también se puede lograr la completa ejecución remota de código, lo que otorga al atacante el control total del servidor en cuestión. En el caso actual, es probable que todas estas posibilidades sean alcanzables. La evaluación de la situación con CVSS 3.1 arroja los siguientes parámetros: AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:H, y una puntuación de 9.9. La razón principal por la que no se alcanza el 10 es que las 24 vulnerabilidades están protegidas por la necesidad de autenticación. En otras palabras, un usuario debe autenticarse antes de poder ejecutar un ataque. Sin embargo, cualquier usuario puede realizar los ataques en cuestión.

El sistema Quipux cuenta con una función de protección contra las inyecciones SQL. Esto se realiza mediante la función `limpiar_sql()`. El principal impedimento de esta función es la imposibilidad de usar comillas simples ('). Si bien esto es suficiente cuando se coloca contenido controlado por el usuario dentro de un elemento de string en la consulta de la base de datos, no tiene mucho efecto en los 24 casos identificados donde es posible una inyección, ya que esta ocurre fuera de un elemento de string.

### **3.1.1 Inyección SQL en la funcionalidad de búsqueda (sql i . 1)**

Esta vulnerabilidad se encuentra en `busqueda/busqueda.php`, línea 86. Ocurre después de la comprobación de autenticación. La inyección se realiza en el parámetro POST `txt_depe_codi` y se ubica al final de la query SQL.

### **3.1.2 Inyección SQL en la funcionalidad de búsqueda 2 (sql i . 2)**

Esta vulnerabilidad se encuentra en `busqueda/busqueda.php`, línea 92. Ocurre después de la comprobación de autenticación. La inyección se realiza en el parámetro POST `txt_usua_codi` y se encuentra al final de la query SQL.

### **3.1.3 Inyección SQL en adjuntos (sql i . 3)**

Esta vulnerabilidad se encuentra en `anexos_lista.php`, línea 66. Ocurre después de la comprobación de autenticación. La inyección se realiza en el parámetro GET `radi_temp` y no se encuentra al final de la query SQL.

### **3.1.4 Inyección SQL en formularios de administración (sql i . 4)**

Esta vulnerabilidad se encuentra en `Administracion/listas/formArea_ajax.php`, línea 25. Ocurre después de la comprobación de autenticación. La inyección se realiza en el parámetro GET `codDepe` y se encuentra al final de la query SQL.

### **3.1.5 Inyección SQL en formularios de administración 2 (sql i . 5)**

Esta vulnerabilidad se encuentra en `Administracion/listas/formDepeHijo_ajax.php`, línea 23. Ocurre después de la comprobación de autenticación. La inyección se realiza en el parámetro GET `codDepe` y no se encuentra al final de la query SQL.

### **3.1.6 Inyección SQL en formularios de administración 3 (sql i . 6)**

Esta vulnerabilidad se encuentra en `Administracion/listas/formDepePadre_ajax.php`, línea 23. Ocurre después de la comprobación de autenticación. La inyección se realiza en el parámetro GET `codInst` y no al final de la query SQL.

### **3.1.7 Inyección SQL en documentos asociados (sql i . 7)**

Esta vulnerabilidad se encuentra en `asociar_documentos/asociar_borrar_referencia.php`, línea 17. Ocurre después de la comprobación de autenticación. La inyección se realiza en el parámetro POST `radi_ume` y al final de la instrucción de actualización SQL.

### **3.1.8 Inyección SQL en documentos asociados 2 (sql i . 8)**

Esta vulnerabilidad se encuentra en `asociar_documentos/asociar_documento_buscar_query.php`, línea 74. Ocurre después de la comprobación de autenticación. La inyección se realiza en el parámetro GET `radi_ume`.

### **3.1.9 Inyección SQL en documentos asociados 3 (sql i . 9)**

Esta vulnerabilidad se encuentra en `asociar_documentos/asociar_documento_grabar.php`, línea 53. Ocurre después de la comprobación de autenticación. La inyección se realiza en el parámetro POST `txt_radi_ume`.

### **3.1.10 Inyección SQL en documentos asociados 4 (sql i . 10)**

Esta vulnerabilidad se encuentra en `asociar_documentos/asociar_documento.php`, línea 33. Ocurre después de la comprobación de autenticación. La inyección se realiza en el parámetro GET `radi_ume`.

### **3.1.11 Inyección SQL en la funcionalidad de ubicación (sql i . 11)**

Esta vulnerabilidad se encuentra en `radicacion/buscar_usuario.php`. Ocurre después de la comprobación de autenticación. La inyección se realiza en el parámetro POST `buscar_tipo`.

### **3.1.12 Inyección SQL en la funcionalidad de ubicación 2 (sql i . 12)**

Esta vulnerabilidad se encuentra en `radicacion/formArea_ajax.php`. Ocurre después de la comprobación de autenticación. La inyección se realiza en el parámetro GET `codDepe`.

### **3.1.13 Inyección SQL en documentos asociados 4 (sql i . 10)**

Esta vulnerabilidad se encuentra en `asociar_documentos/asociar_documento.php`, línea 33. Ocurre después de la comprobación de autenticación. La inyección se realiza en el parámetro GET `radi_num`.

### **3.1.14 Inyección SQL en la funcionalidad de ubicación (sql i . 11)**

Esta vulnerabilidad se encuentra en `radicacion/buscar_usuario.php`. Ocurre después de la comprobación de autenticación. La inyección se realiza en el parámetro POST `buscar_tipo`.

### **3.1.15 Inyección SQL en la funcionalidad de ubicación 2 (sql i . 12)**

Esta vulnerabilidad se encuentra en `radicacion/formArea_ajax.php`. Ocurre después de la comprobación de autenticación. La inyección se realiza en el parámetro GET `codDepe`.

### **3.1.16 Inyección SQL en la funcionalidad de ubicación 3 (sql i . 13)**

Esta vulnerabilidad se encuentra en `radicacion/formDepeHijo_ajax.php`. Ocurre después de la comprobación de autenticación. La inyección se realiza en el parámetro GET `codDepe`.

### **3.1.17 Inyección SQL en la funcionalidad de ubicación 4 (sql i . 14)**

Esta vulnerabilidad se encuentra en `radicacion/formDepePadre_ajax.php`. Ocurre después de la comprobación de autenticación. La inyección se realiza en el parámetro GET `codInst`.

### **3.1.18 Inyección SQL en la funcionalidad de ubicación 5 (sql i . 15)**

Esta vulnerabilidad se encuentra en `radicacion/ver_datos_usuario.php`. Ocurre después de la comprobación de autenticación. La inyección se realiza en el parámetro GET `destinatario`.

### **3.1.19 Inyección SQL en informes (sql i . 16)**

Esta vulnerabilidad se encuentra en `reportes/reporte_TraspasoDocFisico.php`. Ocurre después de la verificación de autenticación. La inyección se realiza en el parámetro GET `verrad`.

**3.1.20 Inyección SQL en transacciones (sql i . 17)**

Esta vulnerabilidad se encuentra en `tx/datos_imprimir_sobre.php`. Ocurre después de la verificación de autenticación. La inyección se realiza en el parámetro `GET txt_usua_codi`.

**3.1.21 Inyección SQL en transacciones 2 (sql i . 18)**

Esta vulnerabilidad se encuentra en `tx/datos_imprimir_sobre.php`. Ocurre después de la verificación de autenticación. La inyección se realiza en el parámetro `GET nume_radi_temp`.

**3.1.22 Inyección SQL en las transacciones 3 (sql i . 19)**

Esta vulnerabilidad se encuentra en `tx/datos_imprimir_sobre.php`. Ocurre después de la verificación de autenticación. La inyección se realiza en el parámetro `GET nume_radi_temp`.

**3.1.23 Inyección SQL en las transacciones 4 (sql i . 20)**

Esta vulnerabilidad se encuentra en `tx/revertir_firma_digital_grabar.php`. Ocurre después de la verificación de autenticación. La inyección se realiza en el parámetro `GET txt_radi_ume`.

**3.1.24 Inyección SQL en las transacciones 5 (sql i . 21)**

Esta vulnerabilidad se encuentra en `tx/tx_borrar_opcion_imp.php`. Ocurre después de la verificación de autenticación. La inyección se realiza en el parámetro `GET codigo_opc`.

**3.1.25 Inyección SQL en las transacciones 6 (sql i . 22)**

Esta vulnerabilidad se encuentra en `tx/tx_realizar_tx.php`. Ocurre después de la verificación de autenticación. La inyección se realiza en el parámetro `GET txt_radicados`.

**3.1.26 Inyección SQL en las transacciones 7 (sql i . 23)**

Esta vulnerabilidad se encuentra en `tx/tx_seguridad_documentos.php`. Ocurre después de la verificación de autenticación. La inyección se realiza en el parámetro `GET txt_radicados`.

### 3.1.27 Inyección SQL en la carga de archivos (sql i . 24)

Esta vulnerabilidad se encuentra en `uploadFiles/cargar_doc_digitalizado_paginador.php`. Ocurre después de la verificación de autenticación. La inyección se realiza en el parámetro `GET txt_depe_codi`.

## 3.2 Cross-site scripting - XSS (xss . 1)

El archivo `anexos/anexos_nuevo.php` contiene una vulnerabilidad de Cross-Site Scripting (XSS). Esta se encuentra en la línea 23, donde el parámetro `POST asociarRad` se inserta directamente en el HTML impreso, sin ninguna limpieza ni verificación de datos. Se puede acceder a esta página sin autenticación. Esta vulnerabilidad está registrada como CVE-2025-55341 y se clasifica como CWE-79. Según CVSS 3.1, sus parámetros son `AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:N`. Según la información disponible, obtiene una puntuación de 6.5.

Esta vulnerabilidad puede ser utilizada por un atacante para diversas acciones, incluyendo el robo de cookies y otros datos privados. También puede utilizarse para ejecutar acciones automáticamente suplantando la identidad del usuario. El hecho de que sea un parámetro `POST` complica estos ataques, pero dado que la aplicación no cuenta con ningún tipo de protección, como tokens CSRF, esta vulnerabilidad se agrava y la convierte en una amenaza para toda la aplicación.

## 3.3 Divulgación de datos (info . 1)

El archivo `Administracion/usuarios/cambiar_password_olvido_validar.php` contiene una vulnerabilidad de divulgación de datos. Se puede acceder a él sin autenticación. Al asignar el parámetro `POST txt_login`, devolverá información sobre la cuenta con el nombre de usuario que coincide con los datos proporcionados. La información revelada incluye la cédula del usuario en cuestión y el tipo de usuario. Para empeorar las cosas, el parámetro `txt_login` realiza una búsqueda mediante el operador SQL "LIKE", que permite el uso de subconjuntos del nombre de usuario. Esto permite enumerar fácilmente la base de datos completa de usuarios.

Esta vulnerabilidad está registrada como CVE-2025-55342 y se clasifica como CWE-200. Utilizando CVSS 3.1, sus parámetros son `AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N`. Según toda la información disponible, obtiene una puntuación de 5.3.

## 4 Recomendaciones

Generalmente, no se considera apropiado intentar limpiar la entrada generada por el usuario para evitar la inyección de SQL. En su lugar, siempre se recomienda utilizar las llamadas queries parametrizadas, que imposibilitan las vulnerabilidades de inyección. La mayoría de los lenguajes de programación son totalmente compatibles con esta función, y PHP no es la excepción.

Para evitar vulnerabilidades de Cross-Site Scripting (XSS), se recomienda limpiar siempre los datos proporcionados por el usuario antes de imprimirlos como parte de páginas HTML. También se recomienda utilizar otras funciones, como las Políticas de Seguridad de Contenido, para evitar la ejecución de JavaScript arbitrario en páginas normales. Esto requiere una separación adecuada entre scripts y páginas HTML. Finalmente, es importante aplicar tokens CSRF y otras contramedidas para reducir el impacto de posibles vulnerabilidades XSS.

En general, conviene definir correctamente las API para acceder a los datos de la cuenta. Estas API deben tener la autenticación y la autorización correctas, y nunca deben exponer más información de la que el usuario debería poder acceder.

Para reducir el impacto de estas vulnerabilidades, es posible utilizar firewalls de aplicaciones web o firewalls de última generación, hasta que sea posible actualizar a la última versión con las correcciones aplicadas.

## **5 Cronología**

### **5.1 Semana del 28 de abril de 2025**

Análisis del código fuente de Quipux realizado por el equipo de Seguridad Digital EC. Se descubrieron y verificaron todas las vulnerabilidades.

### **5.2 3 de mayo de 2025**

Informe inicial enviado a EcuCERT. Solicitud de 3 CVE enviada a Mitre.

### **5.3 9 de mayo de 2025**

Informe inicial enviado a [servicios@gobiernoelectronico.gob.ec](mailto:servicios@gobiernoelectronico.gob.ec). Se recibió una respuesta automática que creó el ticket “2025-14002”.

### **5.4 30 de mayo de 2025**

Carta física al Ministro de Telecomunicaciones ingresada en el Ministerio.

### **5.5 12 de junio de 2025**

Respuesta recibida por parte de Abg. Edgar Roberto Acosta Andrade (Coordinador general jurídico de Mintel).

Correo electrónico enviado al Ing. Cristian Cartuche (Director de Infraestructura, Interoperabilidad, Seguridad de la Información y Registro Civil) por parte de Ola Bini (Analista Principal de Seguridad de Seguridad Digital EC).

## **5.6 16 de junio de 2025**

Respuesta recibida por parte de Ing. Cristian Cartuche, proponiendo reunión el día miércoles 18 de junio a las 3pm.

## **5.7 17 de junio de 2025**

Respuesta enviada a Ing. Cristian Cartuche, proponiendo otras fechas y horas para la reunión.

## **5.8 18 de junio de 2025**

Respuesta recibida por parte de Ing. Cristian Cartuche, proponiendo reunión el día viernes 20 de junio a las 3pm.

Respuesta enviada a Ing. Cristian Cartuche, confirmando fecha y hora de reunión.

Respuesta recibida por parte de Ing. Cristian Cartuche, confirmando fecha y hora de reunión.

## **5.9 20 de junio de 2025**

Notificación completa y detalles de las vulnerabilidades reportadas al Ministerio.

## **5.10 12 de agosto de 2025**

3 CVE reservados.

## **5.11 4 de septiembre de 2025**

Nuevo código con correcciones parciales subida al nuevo repositorio de código fuente.

## **5.12 10 de septiembre de 2025**

Comunicación con el proveedor avisando sobre la divulgación y notificación de que las correcciones no están completas.

### **5.13 16 de septiembre de 2025**

Se recibió una comunicación del proveedor solicitando una reunión sobre el cronograma de divulgación.

### **5.14 17 de septiembre de 2025**

Respuesta al proveedor proponiendo una fecha para la reunión y recordándole el plazo de 90 días acordado.

### **5.15 18 de septiembre de 2025**

Reunión con el subsecretario y otros funcionarios. Se decidió que Seguridad Digital EC trabajará junto con los desarrolladores del ministerio para corregir las vulnerabilidades restantes.

Se extendió el plazo para la publicación. La nueva fecha de publicación es el 1 de noviembre de 2025.

### **5.16 22 de octubre de 2025**

Se realizó una reunión técnica con el Ministerio para revisar y brindar apoyo en la implementación de las correcciones pendientes.

### **5.17 1 de noviembre de 2025**

Este informe fue publicado.

## 6 Referencias

- <https://www.ecucert.gob.ec/>
- <https://minka.gob.ec/mintel/ge/quipux/quipuxcomunitario>
- <https://minka.gob.ec/quipux-comunitario/quipux-comunitario>
- <https://web.gestiondocumental.gob.ec>
- <https://github.com/seguridaddigitalec/quipux-old>