Trabajo Practico

Ejercicio A.1

Investigue 2 ataques informáticos acontecidos en el año 2023, 1 en el sector público y 1 en el privado de su país de origen.

A.1 - Publico

Elemento	Detalles
Fecha del Incidente	25/12/2024 a las 21:30
Organización Afectada	Sitio oficial *argentina.gob.ar* (también Mi Argentina, SUBE)
Equipos involucrados	No especificado; afectación a CMS y páginas estáticas
Volumen afectado	Información visual alterada; sin datos personales comprometidos
Método de Ataque	Intrusion - Ataque de fuerza & Compromiso de la informacion - Modificacion no autorizada de la informacion
Vulnerabilidad explotada	Acceso con clave filtrada a servidor sin segundo factor de autenticación
Identidad del atacante	Seudónimos **h4xx0r1337** y **gov.eth**
Fuente de referencia	Cobertura en Infobae, Perfil, TN, publicaciones técnicas en redes sociales (Capital24, infobae, LinkedIn)

A.1 - Privado

Elemento	Detalles
Fecha	Abril–mayo 2025
Empresa afectada	InformeMédico (software de gestión médica)
Equipos involucrados	No detallado (datos centralizados)
Volumen afectado	665 000+ estudios médicos
Método	Intrusion - Ataque de fuerza bruta & Compromiso de la informacion - Modificacion no autorizada de la informacion

Elemento	Detalles
Vulnerabilidad	Controles de acceso / parches insuficientes
Atacante	No identificado públicamente
Fuente	Informe periodístico (Facundo Quiroga)

Ejercicio A.3

Determine y justifique el impacto en cada pilar en los ejemplos identificados por usted (busque que cada ejemplo valore diferente a los 3+1 pilares). Detalle que información se vio afectada.

A.3 - Publico

En el caso de la pagina de gobierno, se perdio la integridad de la informacion presente en la pagina ya que, la misma debería ser protegida de cualquier cambio no autorizado.

A.3 - Privado

En el caso de InformeMedico se vio afectado la confidencialidad de la información, siendo que los usuarios maliciosos tuvieron accesso a informacion privada de miles de personas.

Ejercicio B

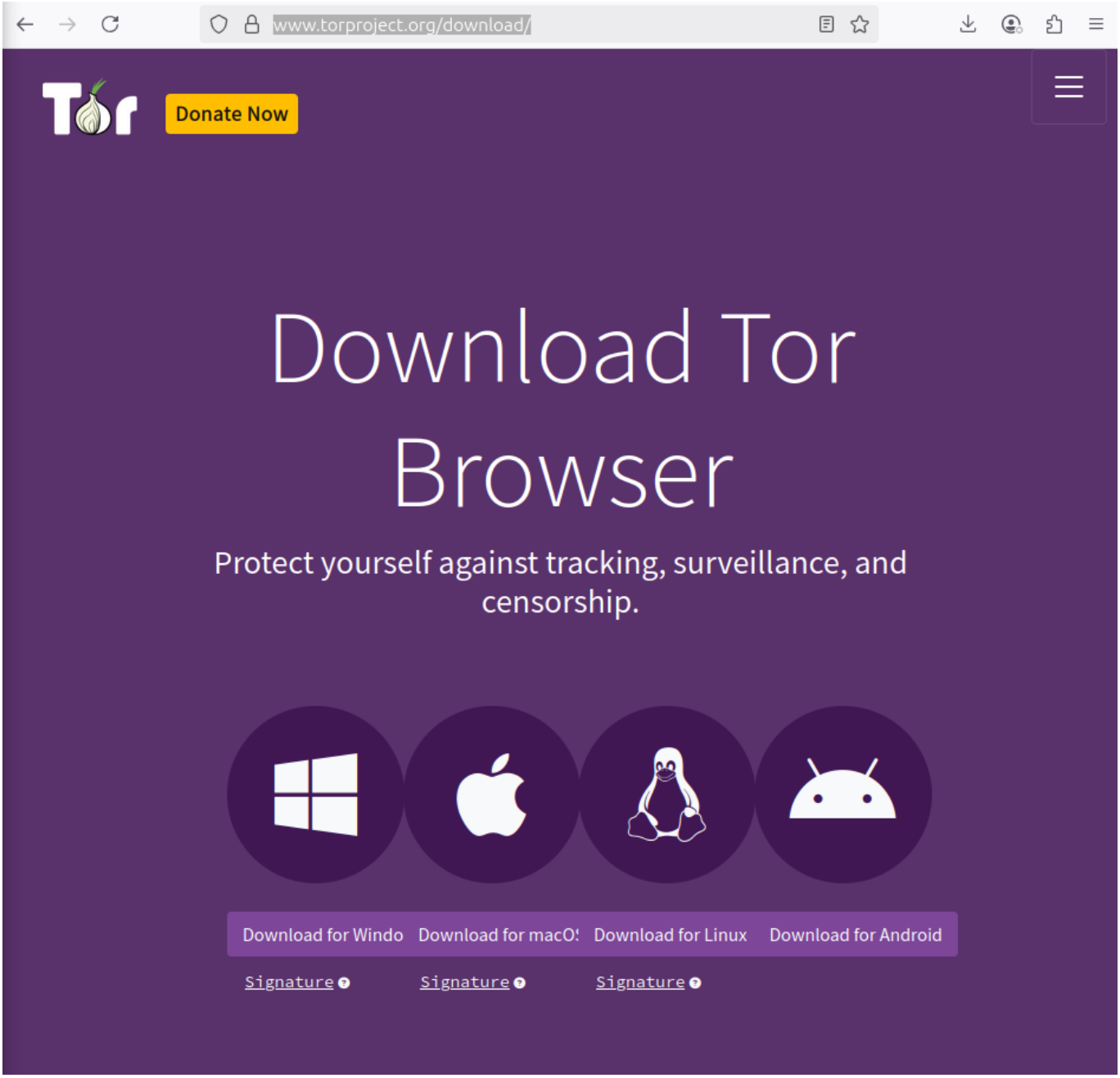
B.1

- Onion BBC News: https://www.bbcnewsd73hkzno2ini43t4gblxvycyac5aw4gnv7t2rccijh7745uqd.onion/
- Clearnet BBC News: https://www.bbc.com/news

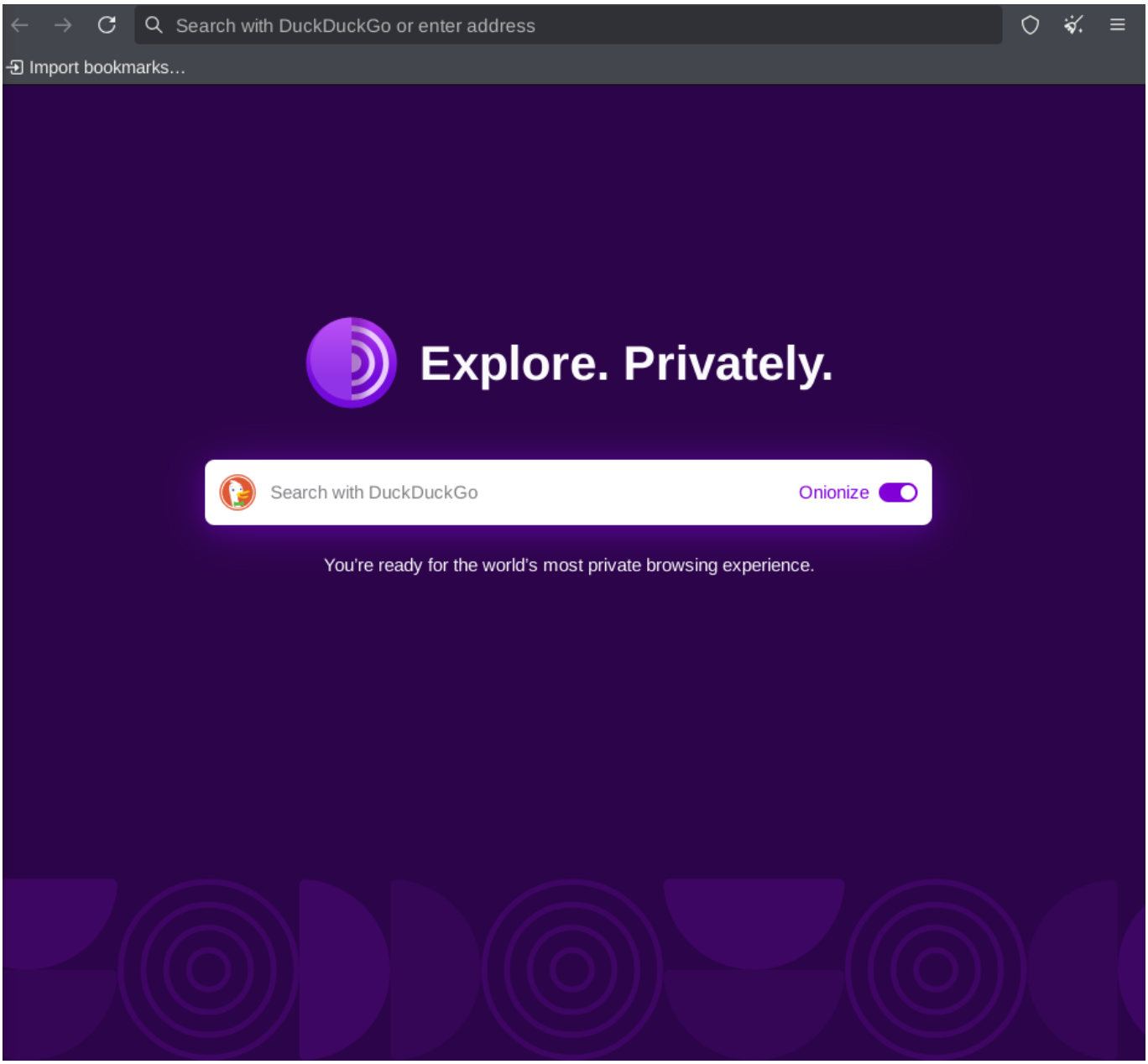
B.2 & B.3

Como usar TOR

Para que un Cliente pueda conectarse a un sitio web utilizando la red tor, primero se necesita de poseer el browser tor ya que, no podemos acceder simplemente a estos dominios que se encuentran en la deep y dark web. Entonces, es necesario hacerse del navegador tor. Para ello, simplemente con ingresar a www.torproject.org, descargar el instalador e instalarlo y ya lo hemos conseguido.



Ya dentro del navegador veremos algo tal que así;



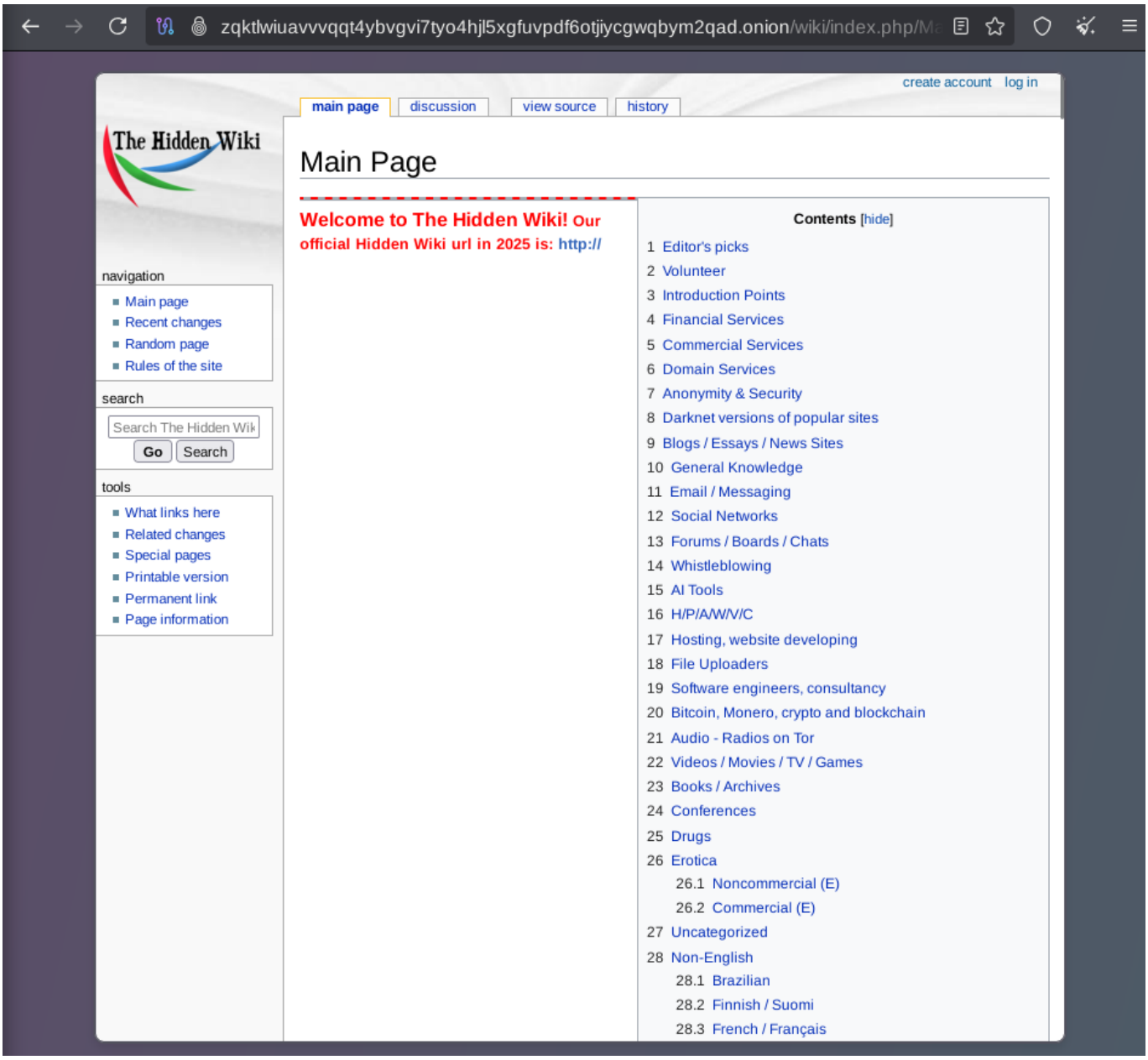
y listo, ya estamos navegando a traves de la red tor y podriamos conectarnos a cualquier sitio web que solemos utilizar.

Que hace (publica) el servidor web

Nada especial, publica DNS, registro A/AAAA, servidor web (HTTP/S) y certificado TLS si aplica. El servidor no sabe ni participa en Tor; desde su punto de vista la petición viene del exit node (y verá la IP del exit).

Como conectarse a un hidden service con tor

Sin embargo, todavia no hemos accedido a un sitio web .onion o tambien llamados hidden services. Para ello, debemos encontrar algun directorio en la clearnet que contenga las urls de los mismos, tal como https://deepweblinks.net/. Aun así, esta pagina solo nos proveera con otro directorio de links pero ahora ya sí, hosteado dentro de la red tor. Uno de los directorios más conocidos dentro de la deepweb es la **Hidden Wiki**.



Y listo, hemos podido conectarnos a un sitio web dentro de la deep web utilizando la red tor.

Que diferencias hay en los servidores web entre la clearnet y la deepweb

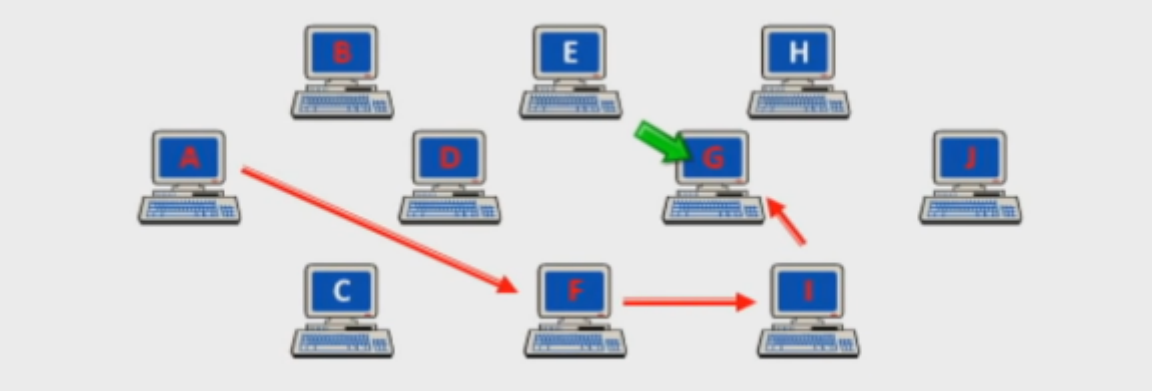
- **No hay exit node:** la conexión entre cliente y servicio .onion se realiza dentro de la red Tor mediante un rendezvous point; no sale a la Internet pública. Por tanto la conexión es end-to-end dentro de Tor. spec.torproject.org+1
- **Dirección = clave pública:** una dirección .onion (v3) se deriva de la clave pública ed25519 del servicio; no depende de DNS público. Onion Services Reddit
- **El servicio se "publica" en HSDirs:** el servidor (hidden service) crea y sube descriptores a unos relays con flag HSDir (Hidden Service Directories) para que los clientes puedan localizarlo. También anuncia introduction points a esos descriptores.

Como funciona la red tor

En el momento que un usuario se descarga tor en su PC, el cliente TOR de dicha maquina obtiene una lista de servidores TOR que se encuentra almacenado en algun servidor, es decir, todas las maquinas que poseen

TOR instalado. Lista que ahora la PC del usuario forma parte al tener TOR instalado.

Luego, haciendo uso de la lista obtenida, el cliente TOR puede acceder a cualquiera (si hostean algun dominio) o hacer uso de ellas como nodos de redireccionamiento. Entonces, si el usuario quiere acceder a una pagina que se encuentra hosteada en una maquina G, el cliente TOR elige una secuencia aleatoria de saltos entre los nodos conocidos para llegar hasta ella.



Primero, antes de enviar la peticion, el cliente se hace de todas las claves publicas de los nodos que va a recorrer. Luego, utilizando la clave publica de G (Nodo destino) encripta la peticion en sí. Despues, el mensaje resultante se le agrega la direccion del nodo G y se encripta con la clave publica de I. Así, hasta llegar al primer nodo. De esta manera, obtenemos un mensaje encriptado en capas donde cada nodo solo puede desencriptar la direccion a la cual debe renviar el mensjae resultante.

Ejercicio C

C.1

sha256sum

- "password" -> 6b3a55e0261b0304143f805a24924d0c1c44524821305f31d9277843b8a10f4e
- "123456" -> e150a1ec81e8e93e1eae2c3a77e66ec6dbd6a3b460f89c1d08aecf422ee401a0

C.2

sha256sum + salt salt = "ABCDEFG"

- "password" + salt -> 3af4e6e1eb0040699029ec748f6c7c89a993093911abfe591958923e3ee1ebf5

- "123456" + salt -> 0acf0e896ea0ed564993b91bd23bc74641d7e502a904b15a0b5d5fc599cc82b1

C.3

hash + salt

- "password" -> yj9T$mlN/lOuxi/W9xGT3Nch9R.$3wD.WZfrYjPOArrHe8PwpW7yf2ktBDk4l2ap/G81dg8
- "123456" -> yj9T$OLYzm6SlmE97BqL4GZEEQ/$8nqsMwU8XM29m33vBkF.P64Dj0SpBL27M5ov5jh1UG/

C.4

Una salt aleatoria rompe patrones, evita que hashes iguales revelen que el dato es igual, y bloquea ataques que usan tablas precalculadas o comparaciones masivas, obligando al atacante a trabajar mucho más y usuario por usuario.