# Diferencias Funcionales entre Cl@ve y Cl@ve 2.0



---

## Autenticación Europea

La segunda versión de éste sistema de autenticación, basado en una nueva especificación técnica y de atributos. Incluye como novedad el acceso al sistema de identificación transfronterizo eIDAS.

## Protocolo

Cl@ve 2.0 emplea un protocolo que no es retrocompatible con las integraciones de Cl@ve, por lo que la transición no puede ser completamente transparente. RedIRIS ofrece mecanismos para maximizar la retrocompatibilidad, como soportar la traducción entre el protocolo de Cl@ve y Cl@ve 2.0. De este modo, a la pasarela de RedIRIS se pueden conectar proveedores de servicio que:

1. Soporten el protocolo de Cl@ve 2.0
2. Soporten el protocolo de Cl@ve
3. Soporten el protocolo SAML2 estándar

El sistema Clave 1.0 empleaba una extensión del protocolo SAML2 desarrollada en el proyecto Stork. Esta variante permite:

- Definir la lista de atributos a pedir en la propia petición (por lo que ésta puede ser dinámica)
- Los atributos de la petición pueden contener valores
- Definir en la propia petición el nivel mínimo de autenticación requerido por el servicio, de modo que sólo se permitirán los mecanismos de autenticación iguales o superiores. Por ejemplo, solicitar un nivel 3 permitirá usar sólo certificados o clave permanente (3) o DNIe (4), pero no clave PIN (2)
- Definir dinámicamente la URL de retorno

Adicionalmente, Clave 1 permitía:

- Fijar un IdP concreto o filtrar algunos (por medio de post params adicionales
- Indicar que se acepten o no certificados de persona jurídica
- Saber si se ha empleado DNIe para autenticar (como atributo en la respuesta)
- Saber si se ha empleado certificado de persona jurídica (como atributo en la respuesta)
- Saber el idp empleado (en el campo issuer de la respuesta)
- Recibir el XML completo de la respuesta de @firma (si se empleó el IDP de @firma)
- Recibir el primer apellido en un atributo separado, para poder separar ambos con seguridad

Clave 2.0 emplea el protocolo eIDAS, también una extensión de SAML2, y evolución del de Stork tras ser adoptado por la Comisión Europea (para eIDAS), pero incompatible, aunque por lo general, equivalente. Este permite lo mismo descrito arriba para stork, pero Clave 2.0 no ofrece tantas cosas como clave 1.0:

- Aún permite fijar un IdP concreto o filtrar algunos (a través de atributos con valor en la petición)
- Aún no implementa la parte de person jurídica de eIDAS (eIDAS define un vocabulario especial para las solicitudes de auth de representación).
- La respuesta de @firma no es completa, se envían sólo algunos nodos concretos, por lo que algún integrador puede perder información
- No permite saber si se ha usado DNIe
- Permite saber el idp empleado, en un atributo devuelto (como RedIRIS implementó en Cl@ve, ya que allí se empleaba un campo reservado en vez de un atributo)

## Endpoints de servicio

En Cl@ve, la pasarela ofrece dos formas distintas de obtener la autenticación Cl@ve. Es decir, soporta dos protocolos:

- **Cl@ve (Stork)**: acepta peticiones en el mismo protocolo que el sistema Cl@ve, aunque emplea un mecanismo más estándar para identificar al servicio.
- **SAML2 Websso estándar**: acepta peticiones en la versión estándar del protocolo, por lo que cualquier SP SAML puede autenticarse en Clave sin esfuerzo de integración. Por contra:
    - La lista de atributos a pedir, el LoA y el acm no pueden ser dinámicos, se definen fijos en los metadatos de la pasarela.
    - Los atributos  no pueden contener valores.

La pasarela Cl@ve 2.0 ofrece tres formas distintas de obtener la autenticación Cl@ve. Es decir, soporta tres protocolos:

- **Cl@ve 2.0 (eIDAS)**: acepta peticiones en el mismo protocolo que el sistema clave, aunque simplifica el proceso eliminando la consulta en vivo de metadatos del SP.
- **SAML2 websso estándar**: al igual que en Cl@ve, con las mismas restricciones.
- **Cl@ve**: La pasarela de RedIRIS sí ofrece retrocompatibilidad a través de este endpoint, que soporta el protocolo de Cl@ve para atacar a Cl@ve 2.0. sólo se necesita cambiar el perfil de atributos solicitados por los de Cl@ve 2.0, que son casi equivalentes, aunque con distinto nombre (más adelante, hay una comparativa). Sin embargo, esta pasarela se recomienda sólo para casos excepcionales o temporales, ya que los kits de integración de Cl@ve hace tiempo que dejaron de tener soporte, y ya entonces tenían graves problemas de seguridad,por lo que se desaconseja su uso.

## Selección del IdP

La pasarela Cl@ve soporta limitar la elección del usuario del IdP a emplear a un subconjunto de los posibles IdPs soportados (o a fijar uno sólo y no visualizar la ventana de selección).

En Cl@ve se realizaba mediante los parámetros POST idpList, forcedIdP e idpExcludedList, donde se especificaba la lista de IdPs a mostrar, a excluir o uno sólo a fijar entre los posibles (aFirma, Stork, SS, AEAT), pero en Cl@ve 2.0 se incluyen como atributos dentro de la petición (uno por cada IdP disponible), de modo que cuando se incluya uno en la petición, este IdP se deshabilitará. Los posibles atributos son:

- **http://es.minhafp.clave/AFirmaIdP**, para desactivar la autenticación con @firma
- **http://es.minhafp.clave/GISSIdP**, para desactivar la autenticación con Cl@ve Permanente
- **http://es.minhafp.clave/AEATIdP**, para desactivar la autenticación con PIN24H
- **http://es.minhafp.clave/EIDASIdP**, para desactivar la autenticación europea

Para los SP que empleen la pasarela WebSSO estándar, ya que el protocolo no permite especificar atributos en la petición, estas opciones de configuración están disponibles en los metadatos de la propia pasarela, pudiendo establecerse por cada SP, pero no dinámicamente por cada petición como puede hacer un SP Clave.

## Autenticación de personas jurídicas

En Cl@ve, algunos IdP de la plataforma permitían que el usuario se autenticase con certificados de persona jurídica. Para ello, junto a la petición se debía enviar en un parámetro POST llamado *allowLegalPerson* el valor 'true'

En Cl@ve 2.0, están soportados los certificados de persona jurídica anteriores a la regulación eIDAS, a extinguir, y no se especifica funcionalidad al respecto, por lo que suponemos que serán aceptados por defecto.

## Solicitud de un nivel de de calidad de la autenticación mínimo

La plataforma Cl@ve soportaba un esquema de QAA para que los SPs solicitasen autenticación con un nivel mínimo, lo que se modelaba utilizando en la petición de autenticación una extensión propia de Stork,, que permitía valores entre 1 y 4:

*<stork:QualityAuthenticationAssuranceLevel>3</stork:QualityAuthenticationAssuranceLevel>*

Para la pasarela WebSSO estándar, esta opción se podía configurar como metadato por cada SP, pero no especificarla dinámicamente en cada petición.

En Cl@ve 2.0 el marco sigue la especificación LoA de eIDAS, de 3 valores (low, substantial, high) que equivale a los niveles 2-4 de QAA y se define en un campo estándar de SAML:

*<saml2p:RequestedAuthnContext Comparison="minimum">*

*<saml2:AuthnContextClassRef>http://eidas.europa.eu/LoA/low</saml2:AuthnContextClassRef>*

*</saml2p:RequestedAuthnContext>*

En este caso, los SP WebSSO estándar pueden elegir enviar el valor o establecerlo en los metadatos.

## Single Logout

[Esta información es provisional, ya que la integración de SLO con Cl@ve 2.0 aún no es funcional y carecemos de la documentación necesaria]

Cl@ve 2.0 soporta peticiones de SLO SP initiated SAML2 estándar, a diferencia de Cl@ve, que introducía cambios no estándar en el protocolo.

Los SP que integren por el endpoint SAML2 WebSSO estándar podrán usar el protocolo de SLO estándar. Para los que integren con Cl@ve 2.0, ofreceremos una interfaz adecuada en cuanto conozcamos la especificación correcta.

## Conjunto de Atributos

A continuación presentamos el perfil de atributos de Cl@ve 2.0:

| Atributo | Tipo | Comentario |
|---|---|---|
| PersonIdentifier | String | Identificador único del usuario autenticado, en la forma:<br>CP/CP/12345678X<br>(CP=Código de país, el primero será el del país de origen del<br>identificador, el segundo el del país de destino)<br>En el caso de identificación de ciudadanos espanoles o extranjeros residentes, el formato será [DNI o NIE], **sin prefijos.** |
| CurrentGivenName /FirstName | String | Nombre propio (El segundo es el friendly name, se envía como nombre en la interfaf SAML2 pura) |
| CurrentFamilyName /FamilyName | String | Apellidos concatenados (El segundo es el friendly name, se envía como nombre en la interfaf SAML2 pura) |
| FirstSurname | String | Sólo el primer apellido |
| SelectedIdP | String | Nombre del IdP que ha elegido el usuario para autenticar ['AFIRMA','PIN24H','SEGSOC','EIDAS']. |
| PartialAfirma | Base 64 string | Si usó el IdP de @firma, devuelve algunos nodos de la respuesta. |
| RelayState | String | Atributo que el SP puede emplear para enviar información de estado en la petición (sólo con el protocolo SAML2-eIDAS) y recibirla de nuevo junto a la respuesta. No confundir con el parámetro POST RelayState, que sí es estándar de SAML2 y también está soportado. |

Este perfil de atributos personales (basado en eIDAS), aunque equivalente, es distinto al empleado en Cl@ve (basado en Stork). A continuación presentamos una tabla de equivalencia:

| Atributo Cl@ve 2.0 | Atributo Cl@ve | Comentario |
|---|---|---|
| PersonIdentifier | eIdentifier | Mismo formato para los identificadores del IDP Europeo (eIDAS), pero para los españoles, se devuelve el DNI sin prefijos |
| CurrentGivenName | givenName | |
| CurrentFamilyName | surname | |
| FirstSurname | inheritedFamilyName,<br>adoptedFamilyName | |
| SelectedIdP | usedIdP | usedIdP fue implementado por la pasarela de RedIRIS, Cl@ve lo devolvía en el campo reservado <issuer>. |
| PartialAfirma | afirmaResponse | La respuesta en Cl@ve era completa, en Cl@ve 2.0, no. |
| RelayState | - | |
| - | eMail | En Cl@ve lo devolvia @firma si el certificado llevaba un correo asociado |
| - | citizenQAAlevel | Ya no se devuelve el nivel efectivo de LoA exacto empleado |
| - | isdnie | |
| - | registerType | |

## Respuesta de @firma

En el caso de que el usuario elija autenticar con @firma, patra ofrecer más información al integrador, se devolvía la respuesta completa de @firma en XML (codificada en base 64). En Cl@ve 2.0, pra reducir el tamaño de la respuesta, se decide devolver sólo los nodos: <dss:Result> y <afxp: ReadableCertificateInfo>.