

# Timestamp Trustly

## Introducción

---

Anteriormente en [Python](#) usamos el concepto de [timestamp](#): simplemente una cadena de caracteres que contiene información de una fecha o tiempo transcurrido de un evento. Es usado en las librerías que manejan fechas `date`.

## Timestamp de Confianza

---

Cuando se crea un archivo o documento y se busca hacer un seguimiento de los cambios y resultados, el timestamp es una herramienta útil. Sin embargo para un estudio/auditoria externo se necesita un mecanismo que asegure que no existan ediciones/modificaciones o eliminaciones (para darlas por inexistentes) de estas timestamps.

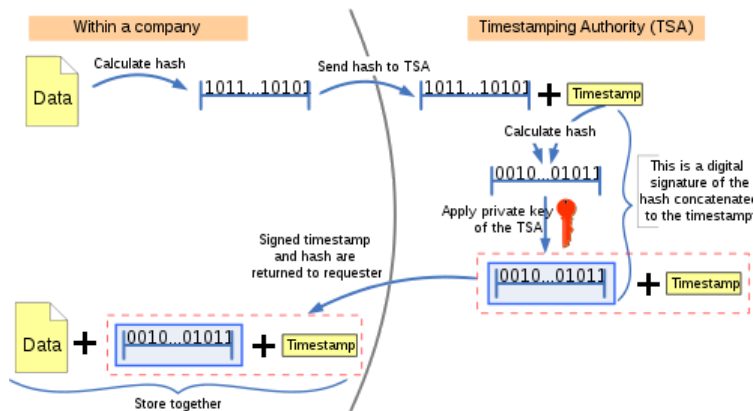
Generalmente los sistemas actuales exigen la creación de firmas digitales juntamente con un timestamp para generar un [Hash](#) y tener una transparencia en el manejo de información pues esta no se puede modificar ni eliminar ni siquiera por el mismo creador/dueño.

## Con un Tercero de Confianza

---

Una forma de llevarlo a cabo es mediante la firma adicional de un tercero de confianza TSA (autoridad de sellado de tiempo).

## Trusted timestamping



Existen empresas privadas que ofrecen el servicio de timestamp de confianza y dependerá de varios factores

## Sin terceros de confianza

Para lograr un timestamp que no dependa de la firma de confianza de un tercero se necesita una referencia que cumpla con ciertas características.

Se debe contar con una fuente de eventos cronológicos, transparentes y públicos que sea inmutable y aproveche los beneficios de un mecanismo de consenso distribuido (resistente a la manipulación y censura).

Desde que llega **Bitcoin** con la generación constante de bloques (con un promedio de 10 minutos por cada uno) es que se propone usar este para generar timestamps de confianza, con empresas como:

1. <https://opentimestamps.org>

Pretende ser un **estándar** de sellado de tiempo para documentos/archivos.

2. <https://stampery.com>

Empresa de certificación basada en Blockchain. Busca aprovechar BC para garantizar existencia, integridad y atribución de datos. Utiliza su propia propuesta que funciona en Bitcoin y Ethereum usando BTA **Stampery Blockchain Timestamping Architecture**

3. <https://proofofexistence.com>

El proceso de acreditación de la existencia de este documento requiere de 3 pasos:

- Cree un hash único que represente el documento.
- Prepare una transacción para escribir el hash en la cadena de bloques.
- Obtenga la tarifa y procese la transacción.

En nuestro proyecto de [Bitcoin/Nodo Bitcoin](#) hemos usado [OpenTimeStamps](#) para certificar binarios y archivos.